Decompilation and copyright in ideas - the protection of non-literal elements of computer software and the idea/expression dichotomy

by

Jacobus Petrus Jooste

Dissertation presented for the degree of
Doctor of Laws in the
Faculty of Law at
Stellenbosch University

Supervisor: Prof Sadulla Karjiker
Co-supervisor: Prof Owen Dean

December 2020

Declaration

By submitting this dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

March 2020

Dedication

To the memory of my father, Kobus Jooste, who was steadfast in his support of my research work and understood the challenges it entails, but did not get to see this thesis finished.

For my mother, Annette Jooste, who is unbearably proud, like only a parent can be, and makes everything seem possible.

Acknowledgements

I am indebted to the Anton Mostert Chair of Intellectual Property Law in the Faculty of Law at Stellenbosch University for providing me with an ideal work and research environment to complete this thesis and for the financial support I received throughout. I am also grateful for additional support from the Faculty of Law and the Department of Mercantile Law, provided to me as doctoral candidate and lecturer.

Sincere thanks to my supervisors, Prof Karjiker and Prof Dean, for their unfailing and patient support throughout. Their insightful, meticulous and expert commentary on my work have made an invaluable contribution.

Thanks also to my family, colleagues and friends for their input and motivation.

Abstract

The maxim, that copyright law does not protect ideas, is frequently challenged when the established principles are tested against new forms of expression or exploitation of a work. The evolution of computer programs, its unique characteristics and the increasing value of software as a commodity have resulted in a strained relationship between copyright law and the public interest regarding access to the underlying ideas in a computer program. This work examines the misalliance between copyright principles and the technical nature of computer programming, with a specific focus on the act of decompiling an existing program where it is undertaken in order to understand the underlying ideas and techniques. The impetus for this analysis is the sui generis classification of computer programs in South African copyright law and the potential this offers for development of domestic law in pursuit of national policy goals. This work conducts a normative analysis of the law and the technical reality of decompilation, from the perspective that copyright law must maintain a clear separation between the idea and the expression. The review of national and foreign copyright law is, throughout, conducted with a perspective on the effect of protection and a critical examination of the degree to which the law maintains an adequate balance between the private and public interests in the protection of software. In this respect, the current legal position is evaluated and a different, normative and pro-developmental perspective regarding decompilation is proposed. It is submitted that a rebalancing of interests is justified and essential in order to establish an appropriate level of fairness and, at the same time, stimulate progress in this industry. It is argued that the act of using computer code to discover its meaning should not amount to infringement in the form of reproduction or adaptation of the work. It is found that the perception of decompilation, as a form of infringement, relies on an analogy to literary work. This view, it is argued, is ill suited to the nature of computer programs, at odds with the sui generis classification in SA copyright law, causes overbroad protection and violates the idea/expression separation. In light of the technical review of decompilation, it is found that the legal basis for prohibiting decompilation as a form of infringement is narrower than commonly assumed and that copyright law principles should be reinterpreted purposefully to permit decompilation. This work advocates that decompilation must be permissible and that an exemption, in SA copyright law, which is limited to decompilation for interoperability alone, is not appropriate in light of the national developmental agenda. Therefore, an alternative exemption is proposed which accommodates the technical reality of decompilation, the public interest in access to ideas and the commercial interests of copyright owners. This approach is supported by an analysis of international copyright law and is based on the inherent flexibilities of the three-step test. The justification for the findings of this work and the proposed departure from foreign precedent is supported by a close examination of the effect of a limited decompilation exception in foreign law and the impact of legislative measures to restrict circumvention of technological protection measures.

Abstrak

Die regspreuk dat outeursreg nie idees beskerm nie word gereeld uitgedaag wanneer die gevestigde beginsels getoets word aan nuwe vorme van uitdrukking of uitbuiting van beskermde werk. Die ontwikkeling van rekenaarprogramme, hul eiesoortige karaktereienskappe en die toenemende waarde van sagteware as ‘n kommoditeit het tot ‘n gespanne verband gelei tussen outeursreg en die openbare belang aangaande toegang tot die onderliggende idees in ‘n rekenaarprogram. Hierdie werk ondersoek die wanverhouding tussen outeursreg-beginsels en die tegniese aard van rekenaarprogrammering met ‘n spesifieke fokus op de-kompilasie van ‘n bestaande program indien sodanige proses aangepak word om die onderliggende idees en tegnieke van die program te verstaan. Die aansporing vir hierdie analise is die sui generis klassifikasie van rekenaarprogramme in Suid Afrikaanse outeursreg en die potensiaal wat dit bied vir die ontwikkeling van plaaslike reg in navolging van nasionale beleidsdoelwitte. Hierdie werk onderneem ‘n normatiewe ontleding van die reg en die tegniese realiteit van de-kompilasie vanuit die perspektief dat outeursreg ‘n beduidende onderskeid moet handhaaf tussen idees en uitdrukkings. Die beoordeling van plaaslike en buitelandse outeursreg word, deurgaans, gedoen met ‘n fokus op die effek van beskerming en ‘n kritiese evaluering van die mate waartoe die reg ‘n behoorlike balans handhaaf tussen die openbare en private belang in die beskerming van sagteware. In hierdie verband word die huidige regsposisie oorweeg en ‘n alternatiewe, normatiewe en pro-ontwikkelingsperspektief voorgestel. Dit word aan die hand gedoen dat ‘n herbalansering van belange geregverdig en noodsaaklik is om ‘n gepaste vlak van billikheid te bereik wat, terselfdertyd, vooruitgang in die industrie kan stimuleer. Die argument word aangebied dat die blote gebruik van programkode, om die betekenis daarvan te ontdek, nie op outeursreg-skending in die vorm van reproduksie of aanwending behoort neer te kom nie. Daar word bevind dat die persepsie van dekompilasie as ‘n vorm van skending steun op ‘n analogie met letterkundige werk. Hierdie beskouing is onvanpas in lig van die aard van rekenaarprogramme, teenstrydig met die sui generis klassifikasie in SA outeursreg, veroorsaak oormatige beskerming en skend die idee/uitdrukking onderskeid. Na aanleiding van die tegniese ontleding van dekompilasie word bevind dat die regsbasis vir die verbod op dekompilasie as ‘n vorm van skending nouer is as wat algemeen aanvaar word en dat outeursreg-beginsels doelgerig heroorweeg moet word om

dekompilasie toe te laat. Hierdie werk voer aan dat dekompilasie toelaatbaar moet wees en dat ‘n uitsondering, in SA outeursreg, wat dekompilasie beperk tot slegs tussenwerking, onvanpas is in die lig van die nasionale ontwikkelingsagenda. Daarom word ‘n alternatiewe uitsondering voorgestel wat die tegniese aard van dekompilasie, die openbare belang in toegang na idees en die kommersiële belange van die outeursreg-eienaar akkommodeer. Hierdie benadering word onderbou deur ‘n analise van internasionale outeursreg en is gebaseer op die inherente buigsaamheid van die drie-stap toets. Die regverdiging vir die bevindings in hierdie werk en die voorgestelde afwyking van buitelandse presedent word ondersteun deur ‘n indringende ondersoek van die effek wat ‘n beperkte dekompilasie uitsondering in buitelandse reg het en die impak van statutêre maatstawwe met betrekking tot omseiling van tegniese beskermingsmaatreëls.

Table of Contents

Chapter 4 The Development of a Decompilation Exception in Foreign Copyright Law and the Role of Anti-circumvention Protection Measures

Chapter 1
Introduction

1 1 Introduction

Much ink has been spilled to explain why, and to what extent, copyright does not protect ideas. The separation of ideas from their expression is at the heart of every copyright discussion because it determines the way in which the law circumscribes the scope of a right and, consequently, the manner and degree to which that right is subject to exception or exploitation.

This has given rise to the idea/expression dichotomy[1] which, as the term suggests, is a quandary – it is short-hand for a question that cannot be answered in the abstract, or in general, in relation to all copyright cases, even where the cases are analogous. It also refers to the need, incumbent on legal development by case law and statutory interpretation, to safeguard the public interest in copyright protection. In this way, the idea/expression dichotomy preserves the flexibility of copyright law to adapt to the variety of expressions and ways of using a work.

Thus, where developments in technology are concerned, it is good that the idea/expression dichotomy remains flexible because it allows copyright law to be tested and, where it is found lacking, to be amended so that it will extend adequate protection to the intellectual endeavour which exist in the expression, but leave the meaning and the message, or any other underlying ideas of the work, free for others to use.

As a balancing mechanism, the dichotomy finds application in two scenarios, namely, the subsistence of protection and the infringement of a right. For example, in the latter scenario, the idea/expression dichotomy is used to argue that reproduction did not

[1] The idea/expression dichotomy traces its origin to the matter of *Baker v Selden* 1879 101 US 99 and has been applied in South Africa since as early as 1920 in *Natal Picture Framing Company Ltd v Levin* 1920 WLD 35 and was fully developed in *Galago Publishers (Pty) Ltd and another v Erasmus* 1989 (1) SA 276 (A).

occur because the contentious work reproduced only the underlying idea from the original, not the expression thereof. In this context, the idea/expression dichotomy is the basis of a defence to infringement.

In the first scenario, the idea/expression dichotomy is used to determine whether an expression qualifies for protection in terms of the act. This asks whether the work, or parts thereof, is the original product of intellectual endeavour or not. In this context, the idea/expression dichotomy is the basis for determining the scope of protection.

In this way, the idea/expression dichotomy attempts to maintain a balance between what is fair to protect and what is fair to leave open for others to put to new use. The essential point is that in every case, there is something that is left unprotected, because it is an idea.

However, in the case of computer programs, there is a situation in which the balancing function of the idea/expression dichotomy is routinely violated – namely decompilation.[2]

The root of this problem has been acknowledged since the earliest days of protecting computer programs in terms of copyright law. As one author puts it:

> "The extension of copyright protection to computer software requires the courts to apply copyright principles to a type of literary work which is different in certain ways from any other protected by the copyright laws. The most fundamental of these differences is that **computer programs lack the communicative function** of traditional literary works. Unlike a novel or a play, where the author's creation is intended to communicate directly with the user, computer programs are designed to communicate with a computer, and only indirectly with the user of the computer.

[2] See Soobert A M "Legitimizing decompilation of computer software under Copyright law: a square peg in search of a square hole" 1994 *John Marshall Law Review* 28 105 at 108 for a description of decompilation. Generally, decompilation refers to a form of reverse engineering of computer code. It is a technical process carried out on the object code, the publicly available version, in order to transform it into source code. Source code is written in programming language and is, therefore, humanly legible. The act, and process, of decompilation is defined in greater detail below in chapter 3.

> Computer program users almost never see the program's underlying source code – the copyrighted work – only its results or output."[3]

This observation is at the core of the research question of this work. If a computer program is made available to the public in a form that does not communicate in human language, the idea/expression dichotomy is *potentially* violated. The program cannot be directly read or understood, which means its ideas are no longer free for others to learn from or use.

Because computer programs are distributed to the public in object code form, which is eminently illegible, the ideas underlying that work are inaccessible. This is not *per se* a problem for copyright law – there is no requirement that a work must communicate with the user, at all, in order to qualify for protection.

However, where copyright law prevents the user from changing, i.e. decompiling, that work into a form that is legible, the idea/expression is *actually* violated. Herein lies the reason why this work focuses on decompilation because, as will become clear, unless copyright law expressly permits the user to take steps to make the work legible, the application of general copyright principles will operate to obstruct it, despite the fact that ideas are said to be excluded from copyright protection.[4]

This is also the reason why the work contributes to the debate on the protection of non-literal elements of software.[5] It does not deal with the question of infringement by

---

[3] Gesmer L T "Developments in the law of computer software copyright infringement" 1986 *Jurimetrics* 26 (3) 224 225 (emphasis added).

[4] See Reed C and Angel J *Computer Law: The Law and Regulation of Information Technology* 6ed (2007) 370 where the authors point out that "copyright law appears to give indirect protection to the ideas underlying a computer program by making the literal copying inherent in simple use of the program an infringing act."

[5] 'Non-literal elements' is a term used to describe any element of a work that is not due to its literal expression and would include, for example, the plot of a book or the thematic elements (or composition) of a cartoon. It is sometimes referred to as the 'look and feel' of a work and, thus, most common in cases involving alleged adaptation of a protected work. In the case of computer programs, the most frequently disputed non-literal elements are the *style, structure and organisation* (SSO) of the lines of code, the manner in which the program adheres to, or implements, functional restrictions to interact

reproducing non-literal elements.[6] It deals with the *de facto* protection that is afforded to everything in the code, including the non-literal elements, by virtue of the prohibition on decompilation. There may be cases where non-literal elements are protectable expressions.[7] However, for the purpose of this work, non-literal elements mean those elements that do not relate to the literal code and are theoretically non-protectable because they represent ideas that should be accessible to everyone.

In other words, 'non-literal elements' are used as a device to refer to those aspects of a computer program, i.e. its ideas, that are useful for others to know and re-use, but which remain inaccessible, when the law prohibits decompilation, or useless, when the law does permit decompilation but only under unduly narrow circumstances.

1 2 The research question

It is in this context that the topic of this work must be viewed in order to identify the scope and focus of this work.

In the first place, it is an examination of the protection of ideas in computer programs regarding decompilation. This is limited to an evaluation of the nature of computer programs and the reasons why decompilation is considered a form of infringement.

---

with hardware, other software or networks, called the *interface specifications*, and the appearance, layout and composition of the visual display, produced by the program, called the *user interface*.

[6] The courts have developed a number of tests to identify which elements are protectable and which are not. See for example the copying and improper-appropriation test derived from the judgment in *Arnstein v Porter* 1946 154 F.2d 464, the extrinsic/intrinsic test based on *Sid & Marty Krofft Television Productions Inc v McDonald's Corporation* 1977 562 F.2d 1157, the total concept and feel test formulated in *Roth Greeting Cards v United Card Company* 1970 429 F.2d, the abstractions test conducted in *Nichols v Universal Pictures Corporation* 1930 45 F.2d 119 and the most common abstraction, filtration and comparison ,or AFC, test in *Computer Associates International Inc v Altai Inc 1992 982 F.2d 693* in response to *Whelan Associates Inc v Jaslow Dental Laboratory Inc* 1986 797 F.2d 1222. See Samuelson P "A Fresh Look at Tests for Nonliteral Copyright Infringement" (2013) 107 *Northwestern University Law Review* 1821 and Velasco J "The Copyrightability of Nonliteral Elements of Computer Programs" (1994) 94 *Columbia Law Review* 242 for an analysis of these tests.

[7] Examples of protected expressions that relate to non-literal elements are provided in chapter 4 below.

In the second place, it is an examination of the impact of decompilation on the idea/expression dichotomy as a balancing mechanism. In other words, it examines the effect of decompilation, and decompilation exceptions, in order to understand why it is not fair to prohibit this practice.

Thus, the core research question may be formulated as follows: **Does copyright law protect ideas because of the way in which decompilation of computer programs is regulated, and is this position fair?**

This leads to a number of subsequent questions which must be answered. First, what is decompilation from a legal and a technical perspective? Second, how does South African copyright law address decompilation? Third, is this position fair and appropriate in light of leading foreign precedent? Fourth, how should copyright law in South Africa be developed to address the idea/expression dichotomy in a fair manner in relation to decompilation and, at the same time, give effect to national developmental goals?

1 3 The methodology

In order to address the above questions, the following must be done.

It is necessary to understand what decompilation is in a technical sense, which requires an illustration of how decompilation is conducted and what happens during this process. In this context, the nature of computer programs as "inherently functional"[8] work is of paramount importance. Thereafter, the legal position in South African (SA) copyright law regarding decompilation must be determined. This leads to a discussion on whether the law accurately reflects the technical reality regarding decompilation, and the focus is on the application of the restricted acts of adaptation and reproduction.

---

[8] Ncube C "Equitable Intellectual Property Protection of Computer Programs in South Africa: Some Proposals for Reform" 2012 *Stellenbosch Law Review* 3 438. See also Tong L "Copyright Protection for Computer Programs in South Africa: Aspects of Sui generis Categorization" 2009 *Journal of World Intellectual Property* 12 (4) 266 272.

Based on the findings up to this point, the question about whether this result is fair will be addressed. This is the fairness analysis, and it is done in two stages.

First, the position regarding decompilation of computer programs in the US and the UK is reviewed to determine how, and to what extent, these jurisdictions achieve a degree of fairness. Based on this analysis, a number of factors, or points of concern, are identified, which remain problematic in light of the idea/expression dichotomy's balancing role. This is not a legal comparative exercise. The purpose is not to compare the laws of the US, UK and SA. The study of foreign law is a normative one. Its purpose is to identify the considerations of fairness that have contributed to the development of a decompilation exception in these countries and draw upon this when an approach for SA legal development is proposed. In addition, the review of foreign law reveals a number of shortcomings which are instructive, and should be avoided, when an approach for SA law reform is suggested.

Second, based on the lessons from foreign legal development on decompilation, an approach to decompilation is drafted. This approach is based on international copyright law and, in particular, the three-step test as the primary regulator of all copyright limitations and exceptions. The purpose of this exercise is to ensure that whatever suggestion is made for SA law, the exception will be justifiable in terms of the three-step test and consistent with international law.

In light of this information, an approach to decompilation is suggested that fits the South African copyright law, is aligned with comparable foreign law on this point and is consistent with international copyright law. The result, it is submitted, is both fair and justifiable. Consequently, it gives meaning to the idea/expression dichotomy in relation to computer programs insofar as it may correct the imbalance between public and private interests, caused by the prohibition on decompilation or an overly narrow exception to decompilation.

1 4 The perspective

The approach of this work to copyright reform is a purposive one. It seeks to address the public/private interest imbalance for a specific reason, namely the developmental objectives of South Africa. In particular, it considers it necessary to conduct research on decompilation, and the consequent protection of ideas, because this has a potentially negative effect on the ability to learn from, and create, new programs. It is submitted that, in the context of a developing economy, it is necessary to make changes to copyright law where it obstructs the establishment of, or growth in, a particular industry. In the SA context, this is the local ICT-based goods and services industry, i.e. the production and distribution of new programs or programs that interact with, or build upon, the work of others.

Therefore, this thesis identifies the developmental objectives and policy decisions that must influence the suggestions made herein regarding the formulation of a decompilation approach. Thereafter, during the fairness analysis, an effort is made to accommodate these objectives in copyright law so that it may influence the nature and scope of a copyright exception that is, nevertheless, aligned with, and justified by, international copyright law.

Thus, this work has a distinctly pro-developmental perspective on copyright reform. However, it is acknowledged from the outset that this may not be allowed to influence the legal findings made in this work unless the impact of the developmental agenda is also justified by copyright principles. As a result, the approach of this work regarding the impact of national developmental goals on copyright law is accommodating but cautious. The relevant South African policy documents are analysed and discussed in chapter 2 below.

1 5 The original impetus for research

This work originated as a response to the discomfort expressed by scholars[9] regarding the classification of computer programs as a *sui generis* type of copyrightable work

[9] These arguments are discussed in more detail in chapter 2 below. See Van der Merwe D *Computers and the Law* 2ed (2000) 78; Pistorius T and Visser C "The Copyright Amendment Act 125 of 1992 and Computer Programs: A Preliminary Overview" 1992 *SA Mercantile Law Journal* 4 346; Tong L "Authorship of Computer Programs under South African Copyright Law" 2005 *SALJ* 122 (3) 513; Simon I "South African Supreme Court Rules on Copyright in Software and Computer-generated Works" 2006 *Journal of Intellectual Property Law and Practice* 1 (11) 696.

and the allegation that this has caused a degree of discord between SA copyright law and that of foreign and international law.[10]

This impetus remains a central part of this work, insofar as it responds to the contention that *sui generis* classification is, on the one hand, problematic for legal development in SA or, on the other, immaterial when it comes to the application of copyright law principles. It is shown that neither contention is correct. In the first place, it is argued that the *sui generis* classification has created an inherent flexibility in SA copyright law that is not only consistent with international law but also express a useful imperative to treat this type of work with the necessary sensitivity as a consequence of its peculiar nature. In the second place, it is shown that the classification of computer programs as literary works is largely responsible for the violation of the idea/expression dichotomy when it comes to decompilation, even in nations where a decompilation exception is granted.

Consequently, the work is critical of the ideological approach to computer programs as a type of literary work. Throughout, this problem is referred to as the literary-works analogy or the literary-analogy – a tendency to make decisions on the nature, scope or infringement, of protection afforded to computer programs by comparing it to an analogous example of literary work.

It is not the intention of this work to cure this error in logic. It is likely impossible to do so, considering that the method of determining the application of copyright law principles by analogy is common to all types of work and effectively maintains legal certainty. To propose a radical departure from the analogy method would be to undermine the application of general copyright law principles.

---

[10] See for example Pistorius and Visser *SA Merc LJ* 346 348-9; The same interpretation has been repeated by others. See De Villiers R "Computer Programs and Copyright: The South African Perspective " 2006 *SALJ* 123 (2) 326; Simon *Journal of Intellectual Property Law and Practice* 696; Visser C "Copyright in Works Created in the Course of Employment: The Supreme Court of Appeal Gives Guidance" 2009 *SA Mercantile Law Journal* 21 591 592.

Therefore, in this work, it suffices to point out the literary-analogy wherever it is evident because, by doing so, it can be shown that the *sui generis* classification has left room to avoid some of the issues created by legal development based on a literary-analogy. This also accommodates the argument that *sui generis* classification may be leveraged to avoid the pitfalls which may occur where careful attention is not paid to the peculiarly technical nature of computer programs when developing copyright law regarding decompilation.

1 6 The construction of legal doctrine

Considering the normative approach to legal development adopted in this work, it follows that this work conceives legal doctrine as an argumentative discipline.[11] This construction has, in particular, the advantage of "putting things into a broader perspective"[12] by allowing research "to take a step back from the interpreted text or any other document"[13] so that "a concrete legal question can be answered, or a case solved, on the basis of generally accepted, or at least acceptable, views."[14]

Because computer programs are unique and useful creations, the above construction is central to the research approach. In order to conduct a review of software copyright law, this work necessarily considers formal copyright law, including case law, and broader contextual materials such as programming techniques and practices, generally accepted observations about the nature of computer programs, the current technological environment and the peculiar characteristics of software usage. In particular, to understand and apply South Africa's developmental objectives regarding copyright law and computer programs, reliance is placed on a number of policy documents to formulate a normative consensus.

---

[11] Van Hoecke M *Legal Doctrine: Which Method(s) for What Kind of Discipline?* (2011) 4. This conception is characterised by the fact that "it is the argumentation to support some legal interpretation or solution that is emphasised, rather than the interpretation as such."

[12] Van Hoecke *Legal Doctrine* 4.

[13] 4.

[14] 4.

However, it is recognised that in legal discourse interpretation and argumentation "appear to be roughly two sides of the same activity, in which interpretation is the goal and argumentation the means for sustaining that interpretation".[15] For this reason, some suggest that legal doctrine is more appropriately labelled a hermeneutic discipline, rather than an argumentative discipline.[16] Be that as it may, this work will not enter this debate beyond this point. It is sufficient to observe that, in some cases, an argumentative construction of legal doctrine is more suitable because it allows for a greater degree of reliance on every-day realities and industry-specific needs. In the case of computer programs in South Africa, the degree of freedom in relation to research materials, afforded by an argumentative construction, is undoubtedly necessary.

While this work takes an argumentative approach, its contribution is to the law and not to technical or legal practice, or merely ideological. In other words, the aim is to bring copyright law closer to the reality it is intended to govern rather than make reality reflect the law. For this reason, it remains a normative study insofar as it reflects legal doctrine as a normative discipline which, in the case of copyright in computer programs, contributes to the balancing of social and economic interests. However, in order to limit the inherent risk of subjectivity[17] that is common to a normative approach, the research presented here is both jurisprudential and practical. This is why reliance is placed on both the technical nature of decompilation and the legal construction of this act in leading foreign and international copyright law.

The search for an "intersubjective consensus"[18] on decompilation is not submitted here as the most suitable approach to the research questions primarily because this work seeks to find a better solution than the prevailing, and divergent, approaches. Furthermore, this work seeks to accommodate *national* policy interests. Therefore, it would be improper to simply rely on the consensus, or majority opinion, about

[15] Van Hoecke *Legal Doctrine* 5.

[16] 5.

[17] 10.

[18] Van Hoecke *Legal Doctrine* 10. An intersubjective consensus is described as the prevailing view of the majority of the most authoritative scholars and jurists on a particular legal doctrinal issue.

decompilation in copyright law because this consensus exists only among foreign scholars and do not accommodate the policy objectives of SA. In addition, where a consensus on the approach to decompilation has been reached in, for example the US, its legal basis is fundamentally different to that of the UK. This makes it impossible to find a consensus upon which SA law reform may be built.

Instead, the normative analysis in this work seeks to identify a *prevailing functional consensus*[19] which would accommodate the consensus on the technical nature of computer programs within the intersubjective consensus on the role and purpose of copyright law in general, and the idea/expression dichotomy in particular.

1 7 The structure of this work

This work consists of six chapters, arranged in order according to the phases of analysis represented by each sub-question of the core research question.

However, the structure of the work is logically systematic, rather than thematically arranged. It is an attempt to direct the enquiry in a technical and legal order and allow the complexity of the analysis to increase steadily. As a result, the solutions to the research questions may be found in several chapters insofar as, in some cases, the solutions become more sophisticated as the work progresses or the analysis of a question depends on the resolution of an earlier problem. However, each chapter has a distinct character and purpose in relation to the research questions and methodology.

Chapter 2 sets out the perspective of this work and addresses the impetus for conducting research in this area of law, outlined above. It examines the developmental objectives in South African law and policy and, thereafter, considers whether the *sui generis* classification has created room for these policy objectives to be applied in copyright reform measures. In the course of doing so, the allegation that SA law is at odds with international law is discussed and dismissed.

---

[19] This term may be viewed as a variation of the intersubjective consensus which incorporates not only consensus on the law but also on national policy and the technical and technological reality.

Chapter 3 considers the technical and legal nature of decompilation, with the aid of an example of code pre and post decompilation, to reveal the extent to which decompilation can be said to reproduce, or otherwise use, object code to deliver source code. This chapter deals exclusively with the position under South African copyright law and examines the exclusive acts of adaptation and reproduction.

Chapter 4 conducts a thorough review of the development of the decompilation exception in the United States and the United Kingdom, respectively. The legislative history, current statutory position, relevant case law on decompilation and the impact of legislative measures to control access to work protected by anti-circumvention technology are discussed, in this order, in relation to each jurisdiction separately.

Chapter 5 conducts a fairness analysis in retrospect, based on the findings in chapters 3 and 4. This chapter starts with a close analysis of the nature, structure and contents of the three-step test and identifies the inherent flexibilities of the test as a tool to accommodate national developmental goals in copyright law. Thereafter, an approach to decompilation in South Africa is formulated and illustrated with the aid of an example of a decompilation exception that achieves a better balance between ideas and expressions and, consequently, a better balance of public and private interests while also giving effect to the developmental goals of South Africa.

Chapter 6 is a brief conclusion and summation of the findings and suggestions made in this work, in relation to each of the research questions.

Chapter 2

The Developmental Approach to Copyright Law Reform in South Africa and the Classification of Computer Programs

2 1 Introduction

2 1 1 The role of copyright in socio-economic progress

The justification for protecting certain expressions of human intellectual endeavour, by means of legal limitations on the free exploitation of the work, stands on three familiar legs, namely, social, cultural and economic impacts.[20] This model for copyright analysis, namely, the ESC model, expresses and addresses the imperative to consider factors such as societal cost and the potential cultural benefits derived from protected works, in addition to the economic impact of legislative and policy intervention.[21] In relation to copyright law, this study subscribes to the ESC impact model because it agrees that a new approach to interpreting the contribution of copyright protection is necessary. In particular, and specifically in the African context,[22]

[20] The purpose of this approach is stated by the World Intellectual Property Organisation (WIPO) as follows:

> "The proposed approach to the creative economy will assist each country in building a reliable foundation upon which an effective copyright/creative economy policy framework can be constructed. The Guidelines are aimed at facilitating the analysis of copyright law, policy and systems in the creative economy and analyzing their relationship to social, cultural and economic outcomes. […] The ESCIA has been developed with the hope that it will expand and enhance comprehension of the operation of copyright law in society. In turn this knowledge can build a greater awareness of the costs and benefits for specific copyright interventions and will, thus, facilitate greater precision and objectivity in the development of laws and policies for creative economies." See WIPO "Draft Guidelines on Assessing the Economic, Social and Cultural Impact of Copyright on the Creative Economy" (2013) 8 (*ESC Guidelines*).

[21] WIPO *ESC Guidelines* 6.

[22] For example, in the African context the impact of a large and established informal trade sector where the financial impact of law or policy is difficult to measure is not reflected in the traditional economic value approach. Further factors, such as the level of employment in the creative industries, the level of access to knowledge and cultural diversity factors are included in the ESC model and play a significantly more important role in the South African context. In the South African context, the difficulties experienced in implementing international copyright law and extracting value from works protected in the developed world, particularly in relation to access to knowledge, are comprehensively addressed in DR Nicholson "Intellectual Property: benefit or burden for Africa?" (2006) 32 *International Federation of*

it is submitted that the traditional approach, which elevates the economic value of protection above the social and cultural factors, does not reflect the full impact of current or proposed policy and legislation or consider the unique attributes of a local market.[23] It is acknowledged that a pro-developmental African context differs from that of developed nations which have a greater capacity to extract value from a regulatory system that is primarily focused on deriving economic value.[24] Thus, a homogenised approach to copyright law, without adequate checks and balances such as the full

Library Associations and Institutions 310. See also DR Nicholson "Free Trade Agreements and TRIPS-plus: implications for developing countries in Africa" (2005) *World Library and Information Congress: 71th IFLA General Conference and Council "Libraries - A voyage of discovery" August 14th - 18th 2005, Oslo, Norway*. See further the authors' argument regarding protecting vulnerable cultural industries in South Africa in T Pistorius & OS Mwim "The impact of digital copyright law and policy on access to knowledge and learning" (2019) 10 *Journal of the Reading Association of South Africa* 1 at 6.

[23] WIPO *ESC Guidelines* 7.

[24] This point is comprehensively addressed by C Ncube "Decolonising Intellectual Property Law in Pursuit of Africa's Development" (2016) 8 *World Intellectual Property Organization Journal* 34 at 37 where the author discusses the position of the African Group regarding the fact that African states often lack the capacity to extract value from instruments such as TRIPS. This point is further discussed in relation to South Africa's ability to implement a fair use system in chapter 5 below. For a comprehensive analysis of the unique ESC factors in the African context, see C Armstrong, J De Beer, D Kawooya, A Prabhala & T Schonwetter (eds.) *Access to knowledge in Africa: The role of copyright* (2010), specifically at para 8.3.1. Other issues such as the lack of centralized and effective licensing organisations in the African context and the delayed dissemination of knowledge through extended periods of protection are raised in L M Palmer "Balancing intellectual property rights with public obligations in developing nations: Lessons from Africa" (2006) 20 *Critical Arts* 62 at 68 and 73. The role of copyright in employment creation, which is both an economic and a socio-cultural factor in the ESC model, and the nature of copyright as public goods, is addressed by H Rønning, P Thomas, KG Tomaselli & R Teer-Tomaselli "Intellectual property rights and the political economy of culture" (2006) 20 *Critical Arts* 1 at 8 and 9. The developmental needs of African states in relation to access to information protected by electronic or technological measures, and the proposal of a functional equivalent for fair use in digital works is discussed by T Pistorius "Copyright in the Information Age: The catch-22 of digital technology" (2006) 20 *Critical Arts* 47 at 54 et seq. Regarding the social and cultural incentives for providing copyright protection, see S Karjiker "Justifications for Copyright: The Moral Justifications" (2013) *South African Intellectual Property Law Journal* 42.

scope of the ESC factors, the risk remains that African countries will remain incapable of "constructing a responsive copyright regime"[25] that favours access to knowledge.

In addition, the ESC model encompasses a large volume of work on the justifications for copyright protection and presents a constructive interpretation of the debate from which this study may draw, without the need to repeat or summarise trite arguments, and facilitates a critical construction of copyright in computer programs from a socio-economic and cultural perspective.

Furthermore, the impact assessment guidelines developed by the World Intellectual Property Organisation (WIPO) for the ESC model, the ESCIA guidelines, encompass the full spectrum of evaluative measures "for assessing the ways in which copyright law and policy interventions impact economic, social and cultural outcomes in society, as well as the **objectives of national governments.**"[26] The ESCIA guidelines are themselves based on comprehensive research[27] and strong methodology[28] about the impact of economic, social and cultural rights or factors on the creative industries, particularly copyright, and, therefore provide support for a comprehensive balancing of interests within the context of a particular nation and a particular work. As such, it emphasises the flexibilities of the copyright system, provided that it is used in an evidence-based manner, rather than responses to general policy ideals.

WIPO describe nine main ESC indicators to consider when assessing the impact of law and policy in the copyright-based industry. As such, these factors are, for the purpose of this work, useful when considering the impact of ESC rights on copyright reform, particularly where the reform is in pursuit of a national objective. Each of the nine indicators are divided into a range of specific sub-factors.

---

[25] ES Nwauche "The public interest in Namibian copyright law" (2009) 1 *Namibian Law Journal* 57 at 58.

[26] WIPO *ESC Guidelines* 8 (emphasis added).

[27] WIPO Office of Strategic Use of Intellectual Property for Development 2007 *Study on the Economic, Social and Cultural Impact of Intellectual Property in the Creative Industries: Final Report*.

[28] Regarding internal methodology see WIPO *Evaluation Policy 2016-2020 IOD/EP/2016* and regarding procedure see WIPO *Evaluation Manual IOD/EM/2019*. Regarding methodology applied to the ESCIA, see WIPO *ESC Guidelines* 110 *et seq*.

These indicators and sub-factors are discussed in great detail in the ESC guidelines, and may be summarised as follows:

2 1 1 1 *Economic indicators*

The creative economy output[29], the national share of trade in copyright goods and services[30] and finance for, and investment in, the industry.[31]

2 1 1 2 *Social indicators*

The employment in the creative industries,[32] access to knowledge and education[33] and the use of information and communications technology.[34]

2 1 1 3 *Cultural indicators*

The effectiveness of the regulatory framework for cultural development through the

---

[29] The quantity, quality and value of the output of copyright works in the national economy, the current and potential employment creation attributable to the creative economy, the contribution it makes to the GDP and the rate and volume of consumption of copyright work. See WIPO *ESC Guidelines* at 59.

[30] Volume of exports and imports of creative goods and services, the trade deficit or surplus, the competitiveness in a global or regional market, access to, availability of and diversity in creative goods and services, trade and investment flow, tariffs, the percentage attributable to e-commerce, the effectiveness of the legal framework to facilitate trade, the rate of production of local goods or services and the local enforcement of rights. See WIPO *ESC Guidelines* at 64.

[31] Including direct foreign investment, tax and fiscal incentives, public investment, subsidies and private investment. See WIPO *ESC Guidelines* at 69.

[32] Including access to employment, the existence and efficiency of professional organizations, the size of the market, poverty, social security, income inequality, social status, minority participation, and the levels of income in the industry. See WIPO *ESC Guidelines* at 78.

[33] Among other factors the availability of copyright instruments, licensing schemes, limitations and exceptions. The legal and technical barriers to access. The nature of public policy on freedom of expression, diffusion of knowledge, social and cultural identity, social cohesion and diversity. The range of distribution channels and diversity of published works and the availability of published works, e-publishing and levels of skilled professionals. See WIPO *ESC Guidelines* at 83.

[34] In particular the level of access to digital resources and the quality, geographic reach and production output of ICT-based goods and services. Use of ICT-based works in the creative sectors, software expertise and training and the use of ICT for social sharing of works. See WIPO *ESC Guidelines* at 86.

creative economy,[35] the relevant infrastructure[36] and cultural representation and diversity.[37]

A review of these factors makes it clear that the ESC model recognises an inter-relationship between the economic, cultural and social impact factors but maintains the need for each to be assessed and implemented separately. As distinct, yet related, criteria, the factors described in the ESC model can, and should, be applied *individually* – as opposed to the general socio-economic criteria more commonly used in copyright jurisprudence as a surface standard for the balancing of private and public interests and/or the appropriateness of a particular fairness mechanism.

In light of the fact that this work conducts a re-interpretation of copyright regarding decompilation of computer programs only, and seeks to rebalance the interests by *internal* measures, i.e. internal to the nature and function of the type of work rather than external in relation to general copyright principles, it is necessary that a new model for balancing interests be adopted that is capable of distinct application.

It is submitted that the breadth and scope of the ESC impact model allows for consideration of social, cultural and economic impact factors in a normative yet individualised manner to, first, the particular type of work and its peculiarities and, second, the antecedent principles of copyright law according to the revised approach

[35] Including tax and fiscal incentives for activities, the tax contribution of cultural and creative economy small to medium enterprises (SMEs), the sustainability of copyright-based businesses, the availability of creative content/cultural diversity, equitable remuneration to creators, adherence to the rule of law, the investment climate, the cultural capital, regulatory knowledge, the number and ease of access to public institutions and government subsidized cultural programmes. See WIPO *ESC Guidelines* at 95.

[36] In particular the existence of institutions and mechanisms supporting creative communities, the relevant cultural and creative economy policy, availability of creative content, political commitment, social equity, national policy formulation, cost of access to creative output, scope and coverage of available works and the legal and technical barriers to access. See WIPO *ESC Guidelines* at 100.

[37] Among others copyright facilitated native cultural production and cultural preservation. The value attached to national culture by society and national cultural appreciation. Copyright awareness and access to institutions among minority groups. Ethnic and linguistic diversity in copyright-based production. Government subsidies for diversity and the share of minority groups in cultural production. Barriers to cultural participation of minority groups. See WIPO *ESC Guidelines* at 105.

supplied below. In other words, where this work relies on the idea/expression dichotomy as a short-hand term to describe the need for a balance in copyright law, it relies on the ESC model to inform that doctrine.

This study is foremost a construction of copyright in computer programs *in South Africa* which seeks to rebalance the private and public interest in this type of work, without the need to reconsider the impact of its findings on other types of work. In other words, it seeks to circumscribe the current status of computer programs in isolation of other types of work, but within the existing principles of copyright law, with a particular goal in mind. This goal is to establish an understanding of the primarily utilitarian nature of computer programs in terms of the Copyright Act,[38] and amend the most pressing misalignment, namely, access to code, accordingly. For these reasons, it is necessary to review and summarise, further below, the South African context and identify the ESC factors as expressed in national policy.

In addition, it is considered fundamental that a contemporary reworking of copyright in computer programs, in South Africa, should advance the debate on the impact of copyright as a stimulus for social, cultural and economic progress. It is no longer acceptable to merely bemoan the perceived imbalance between private and public interests, or to circumscribe the traditional justifications for copyright within a global contextual reading of copyright in relation to all types of work.

It is, instead, necessary to address the specifics of copyright protection and its effect in relation to specific types of work, in light of all of the ESC factors within a national context, and make radical changes to the law if such amendments are justified. There may not be cause to conduct such a construction in isolation for each type of work, but it is submitted that, at least for computer programs, the type of work should be evaluated specifically, for two reasons. First, in SA computer programs are protected as a unique type of work and, second, this type of work has a peculiar social, economic and cultural impact potential.

---

[38] Copyright Act 98 of 1978 (Copyright Act), as updated to *Government Gazette* No 32121 (9 April 2009).

Any construction of copyright law, and a utilitarian reconstruction in particular, must, however, have as its foundation a firm understanding of the purpose of copyright law. Failure to do so would result in suggestions that are either unworkable, because it relies on further judicial/legislative interpretation, or suggest a construction of copyright in computer programs that is neither suited to the socio-economic conditions in South Africa or the needs of stakeholders in this type of work. For these reasons, the ESC impact model guides this study in principle and is drawn upon, by referring to the provisions in international law that give effect to the ESC model, to support its findings. The nature of the ESC model, as it is enshrined in international copyright law, is discussed and applied in detail below in chapter 5.

The advantage of positioning this study, which is in essence a utilitarian, technically specific analysis of one type of commercially-significant copyrightable work, within the ESC model for copyright analysis, is threefold: it ensures that this work remains alert to the full spectrum of interests in copyright protection; it avoids an over-reliance on the economic justifications and facilitates findings that depart from the assumption that private and public interests must be maintained in equal balance in the case of all copyrightable types of work[39] and; it places greater emphasis on the utilitarian nature of the work by accepting that copyright law should not only address the utility of the work within the law but also address the utility of the work itself in light of the social, cultural and economic context.[40]

It is not the purpose of this work to evaluate the motivations for copyright protection or to elevate one above the other. However, it is submitted that the impetus for vesting copyright in any type of work is, at least in part or at the very least indirectly, due to all three reasons, namely, social, cultural and economic considerations. For example, the author of a computer program may seek to address his personal financial situation by creating a new application aimed at facilitating the distribution of digital artwork. In this situation, copyright law operates to reward the programmer for his intellectual

[39] This is done, in particular, by critically analysing the literary-work analogy. For an overview of the problems associated with the literary-analogy, in the case of computer programs, see paragraph 2 2 4 below.

[40] This requires close analysis of case law, conducted below in chapter 4.

endeavours, it incentivises the creation of further works by allowing others to build upon the existing technology and it invests in the creation of further means to serve society. In the latter case, it must be noted that any pecuniary advantage that adheres to the author, as a result of the exploitation of his copyright, may be viewed simultaneously as an economic and social justification for copyright. The author is enriched, which allows him to expend further intellectual effort at creation (the economic justification),[41] but it also allows him to better the socio-economic circumstances of those under his care (the social justification).[42] Furthermore, the creation of new wealth through copyright, and the subsequent distribution of that wealth, expands and grows the collective knowledge (the cultural justification) and increases the potential economic return.

This, admittedly, oversimplified view of copyright, conveys the core of the traditional view of copyright.

It is submitted that the justifications for copyright protection may be collectively viewed, for the purpose of this study, according to its ultimate common result, namely, *progress*. The concept of progress in the context of intellectual property, and copyright in particular, refers to the acceptance of knowledge creation as a disruptive process based on modification of existing work and ideas.[43] In other words, to view the justifications of copyright protection as an essentially progressive tool, emphasis is placed on its ability to reduce the risk of failure when creating new work by relying on established, vindicated and tested works.[44]

---

[41] See Karjiker S "Justifications for Copyright: The Economic Justification" 2014 *South African Intellectual Property Law Journal* (2) 13 for a detailed analysis of the economic justifications.

[42] Karjiker S "Justifications for Copyright: The Moral Justifications" 2013 *South African Intellectual Property Law Journal* (1) 42 where the social and cultural justifications are discussed as part of a general, moral justification.

[43] See Van Caenegem W *Intellectual Property Law and Innovation* (2007) 2 where the author explains that progress, in this context "rejects a world view which prioritises stability" and emphasises that "knowledge is not simply accumulated for its own sake, but with a view to applying it to practical ends".

[44] Van Caenegem *Intellectual Property Law and Innovation* 3.

The limited exclusivity established by copyright protection, in favour of its creator, is a necessary and indispensable first step to economic and social progress. Thereafter, the application of the rights that flow from protection, must, it is submitted, continue to serve both economic and social progress from the perspective of parties other than the creator of the work. This fact is made clear in the diversity, and combination, of impact factors that make up the ESC model. This suggests that the construction of copyright in computer programs as a mechanism for incentivising the creation of new work, requires that the work be made available to the public in a manner that is *useful*. In this sense, a program is useful when it can be read, studied, analysed and interpreted freely.

Thus, the concept of progress, described above, is present throughout this work. In this chapter, progress is used to identify and describe the national policy objectives regarding development. In chapter 3, the concept of progress underlies the examination of computer programming practices and, in particular, the role of decompilation as a tool for the creation of new work. In chapter 4, the development of software reverse-engineering exceptions in foreign law is examined, in order to identify how, and to what extent, the limitations of these exceptions hamper progress. In chapter 5, the concept of progress developed throughout the preceding chapters, is used to test the extent to which international copyright law provides flexibility for the incorporation of new exceptions which seek to advance a developmental goal.

2 1 2 The decompilation of computer programs in context

It is in this context that the decompilation prohibition finds application to the research conducted in this work. It serves to initiate the reconstruction of copyright in computer programs in a South African context and with a socio-economically balanced outcome in mind. In other words, the *perceived* prohibition on decompilation is both the problem under discussion, and the primary symptom which this work seeks to address. Furthermore, the decompilation prohibition delineates the discourse by restricting the study to copyright in one type of work, namely computer programs.

This is not merely a demarcation exercise. The focus is placed on computer programs specifically in order to illustrate that this type of work should not be treated as analogous to, or made subject to, the same copyright principles applicable to other types of work. Instead, it is proposed that the nature of computer programs necessitate a nuanced application of copyright law which may, and in some cases should, depart from existing jurisprudence. Furthermore, it is argued that, in light of the socio-economic considerations of copyright, the nature and scope of copyright in computer programs *in South Africa*, must depart from the general norms and principles of copyright espoused in foreign law. As will become clear, it is found that such a departure need not be drastic, or cause disharmony with international copyright law, in order to achieve local policy goals.

It is suggested that South Africa is in the unique position, directly and indirectly, as a result of the 1992 Copyright Amendment Act,[45] to achieve a sound balance between the private and public interests in copyright, at least insofar as computer programs are concerned. However, this advantage has, thus far, not been explored and has been, to a large extent, counteracted by interpretations of copyright law that are, at best, impractical in a developing economy and, in the worst cases, counter-productive to national development. In particular, the development of copyright in computer programs has bound this type of work to a body of copyright law to which it does not, and should not belong, namely, literary works.[46] As a result, the benefits which may be derived from the unique treatment of computer programs in South Africa have failed to materialise and, in addition thereto, imported several doctrinal dilemmas which should never have occurred, namely, an overly restrictive approach to the protection of code which violates the idea/expression dichotomy,[47] the inappropriately wide

[45] The Copyright Amendment Act 125 of 1992, published in the *Government Gazette* No 14129 Vol 325 on 10 July 1992 (the 1992 Amendment Act).

[46] See Cohen J E “Reverse Engineering and The Rise of Electronic Vigilantism: Intellectual Property Implications of “Lock-Out” Programs” 1995 *Southern California Law Review* 68 1091 at 1107 where the author submits: “To the extent that generalizations about the nature of “literary works” are possible, however, what they reveal is that the statutory classification of computer programs as literary works confuses more often than it clarifies.” This point is argued in more detail below in chapter 3 in relation to the technical nature of decompilation and further in chapter 4 regarding the development of a decompilation exception.

[47] See Van Caenegem *Intellectual Property Law and Innovation* 183 where the author points out that the regulation of decompilation by copyright law necessarily involves a theoretical exploration of the idea/expression dichotomy because the law is “constraining,” from a progressivist point of view, in this respect.

application of anti-circumvention measures to programs and the erosion of fair dealing as a balancing mechanism.

2 1 3 Copyright and South Africa's developmental approach to legislative reform

When copyright protection is considered as an impetus for progress, it becomes essential that the law should be applied in a manner that, at the very least, does not inhibit progress and promotes a "vibrant commons"[48] upon which further developments may be built. Thus, it is necessary that the application of copyright law should seek to balance the interests of the copyright owner and the aspiration of society in order to give effect to the purpose of protection, i.e. the advancement of beneficial intellectual endeavour.[49]

This need to maintain a balance between the private and public interests in copyright is well established as the justification for fair dealing exceptions to the restricted acts. It is expressed as a general guiding principle, incumbent on South Africa as a member of the World Trade Organisation (WTO) and signatory to TRIPS,[50] as follows:

> "The protection and enforcement of intellectual property rights should contribute to the **promotion of technological innovation** and to the **transfer and dissemination of technology**, to the mutual advantage of producers and users of technological knowledge and in a manner **conducive to social and economic welfare**, and to a balance of rights and obligations."[51]

---

[48] C Ncube "Equitable Intellectual Property Protection of Computer Programs in South Africa: Some Proposals For Reform" (2012) 3 *Stell LR* 438 449.

[49] Some express this concept by referring to the characteristic of copyright protection as, inter alia, a "public good," which is served when the information conveyed by the work is allowed to flow freely. See Wang J *Conceptualizing Copyright Exceptions in China and South Africa* (2018) 15 for an overview of sources on this point. This view, although not inaccurate, is not helpful when formulating an approach to copyright exceptions and limitations, because it relies too heavily on the distinction between public and private property, rather than the balance of fairness regarding access to information. Wang does acknowledge this point of critique, later in the work, at 35 *et seq*. A more comprehensive analysis of the public/private property argument in copyright is conducted in Druey J N "Information Cannot Be Owned: There is More of a Difference than Many Think" 2004 *The Berkman Centre for Internet and Society, Harvard Law School Research Publication* 4 1 para 3 *et seq*.

[50] Agreement on Trade Related Aspects of Intellectual Property Rights (TRIPS), annex 1C of the Marrakesh Agreement Establishing the World Trade Organization 15 April 1994.

[51] TRIPS Article 7 (emphasis added).

In addition, TRIPS provides that member states may adopt legislative measures "to promote the public interest in sectors of vital importance to their socio-economic and technological development"[52] and may need to take steps to prevent users of protected works from resorting to "practices which unreasonably restrain trade or adversely affect the international transfer of technology."[53]

These guiding principles are elevated to a positive duty incumbent upon South Africa as a state party to the International Covenant on Economic, Social and Cultural Rights.[54] Article 15 of the ICESCR declares that nationals are entitled to "enjoy the benefits of scientific progress and its applications"[55] and "to benefit from the protection of the moral and material interests resulting from any scientific, literary or artistic production of which he is the author."[56] However, the protection and promotion of these rights are made subject to the following proviso:

> "The steps to be taken by the States Parties to the present Covenant to achieve the full realization of this right [in article 15] **shall include those necessary for the conservation, the development and the diffusion of science and culture**."[57]

Therefore, the duty to facilitate the enjoyment of intellectual endeavour by providing legal protection for the rights of the author is, expressly, made subject to the imposition of measures for the diffusion of the knowledge created by a proprietary regime for the purpose of development in science and culture. Consequently, it is imperative that the rights that flow from authorship, i.e. copyright, should be accompanied by positive measures for the dissemination of the work.

---

[52] Article 8(1).

[53] Article 8(2).

[54] International Covenant on Economic, Social and Cultural Rights, adopted and opened for signature, ratification and accession by General Assembly resolution 2200A (XXI) of 16 December 1966 entry into force 3 January 1976, in accordance with article 27 (hereinafter ICESCR).

[55] ICESR Article 15(1)(b).

[56] Article 15(1)(c).

[57] Article 15(2) (emphasis added).

In addition, when developing measures, a "normative interpretation"[58] of the rights should be applied. Article 15(2) *obliges* state parties to implement progressive measures for the diffusion of work for the purpose of development as part of the steps it takes to vest protection in authorial works. The ICESCR thus places an intra-copyright limitation on the manner in which state parties apply this right. This means that the nature and scope of copyright should be interpreted in favour of dissemination, at least insofar as the balancing of private and public interests are concerned. It also means that copyright limitations should not be sought, in the first place, outside of the Copyright Act or in alternative practices such as open access or copyleft. Instead, the deeming limitation in the ICESCR facilitates a construction of copyright law that is self-limiting and pro-developmental, based on a normative approach.

In South Africa, these sentiments are included as part of the primary objectives of the anticipated national policy on IP law.[59] The DNPIP aims to "promote research, development and innovation throughout the South African economy"[60] by, inter alia, applying a strategy "aimed at building domestic capacity and skills, enabling stakeholders (industry and academics, but also the general public) to better absorb knowledge and use it in their particular environment."[61]

In the case of copyright, it must be noted that the sentiments expressed above are not general or unsupported. The socio-economic impact of the creative industries, and the contribution of copyright-based endeavour, is "substantial enough to stipulate increased attention by the South African policy-makers."[62] Figures published by WIPO

---

[58] PWESCR (Programme on Women's Economic, Social and Cultural Rights) *International Covenant on Economic, Social and Cultural Rights: A Handbook* (2015) 29.

[59] Draft National Policy on Intellectual Property, 2013 General Notice No 918 of 2013, *Government Gazette* No 36816 of 4 September 2013 (DNPIP). The DNPIP was followed by the first implementation instrument, the Intellectual Property Policy Of The Republic Of South Africa Phase I 2018, *Government Gazette* No 41870 of 31 August 2018. The phase 1 policy deals only with intellectual property and public health in the patent law context, and international cooperation arrangements, which fall outside the scope of this work.

[60] *DNPIP* Objective 8.

[61] Objective 15.

[62] Pouris A & Inglezi-Lotz R *The Economic Contribution of Copyright-Based Industries in South Africa* (2011) 5.

from a decade ago show that the copyright-based industries contributed 4.5% to the GDP of South Africa and 4.08% to employment.[63] It should be noted that the contribution of the copyright-based industries have remained stable at between 4% and 4.5% since the initial rise ending in the 1980's.[64] However, although the copyright-based industries contribute to the economy in the form of exports (2.77%) this is outweighed by the total imports at 7.85%.

This observation is of crucial importance when copyright law in South Africa is applied or developed in order to remain in step with foreign law. The socio-economic benefits that may be derived from copyright are not necessarily, or usually, served by a strict regulatory regime. In fact, the developmental agenda of South Africa militates against the increasing, and costly, imposition of copyright enforcement measures.

2 1 4 Cumulative innovation and national development

In the case of computer programs, this point is illustrated by the "cumulative innovation"[65] principle which indicates that the software industry relies, to a great extent, on extracting value from incrementally-improved works created by "a process of sequential development".[66] This process is based on the creation of derivative works in the broad sense.[67] A derivative computer program may perform the same or similar function in a more efficient manner, or perform the function in a different digital environment, or perform an entirely different function based on the same or similar data processing or calculation methodology. For all three reasons, reverse engineering of software is an essential step in learning from existing work in order to build upon it. In programming practice "it would be fair to say that […] reverse engineering for the purpose of developing competing products is the most well-known

---

[63] Pours & Inglezi-Lotz *Economic Contribution* 6.

[64] 6.

[65] Soobert *John Marshall Law Review* 116 n61 and 117 n68, with reference to Victor D "An Analysis of an Affirmative Defense for Reverse Engineering Within A System of Legal Protection for Computer Software" 1993 *Southern California Law Review* 66 1705.

[66] Soobert *John Marshall Law Review* 117.

[67] In this context, a derivative work is a program that takes inspiration from another program, but does not amount to an adaptation of an existing computer program.

application of reverse engineering,"[68] but it is not the only legitimate reason why decompilation may be necessary.[69]

By consulting the code of an existing program, the author is able to re-apply the initial investment in research and development (of the code of first instance) to derive an income from new derivative works which, as a matter of course, can be developed faster and at a lower investment cost. This in turn allows the works to be made available at a lower cost to the consumer, which stimulates and diversifies the market, in both the quantity and quality of programs created. The net result, it is submitted, is a double positive (a benefit to creator and the consumer) consonant with the ideal model for progress.[70] This model, called the "virtuous circle",[71] describes the ideals of a "formal social compact"[72] that endorses the three tiers of progress, namely "growth, development and nation building."[73]

---

[68] Eilam E (2005) *Reversing Secrets of Reverse Engineering* 4.

[69] Decompilation is frequently necessary to create a replacement platform or program in order to provide the same functionality of a program that has become obsolete, is no longer commercially available or no longer supported by updates. In other cases, decompilation plays an important part in software and network security research and maintenance, particularly in the case of encryption research or analysis of the program to detect vulnerabilities to malware. Decompilation is also an essential part of software maintenance practices such as code unification, to create a unified approach or language, and migration of code from one environment to another while maintaining consistency. In addition, an array of "pedagogical uses" of decompilation and the code created by decompilation may justify this act. See further Burleson D "Have decompilers become evil?" Burleson Consulting http://www.dba-oracle.com/t_decompilers_evil.htm (Last Accessed November 2019). See also Cohen J E "Reverse Engineering and The Rise of Electronic Vigilantism: Intellectual Property Implications of "Lock-Out" Programs" 1995 *Southern California Law Review* 68 1091 at 1120.

[70] The National Development Plan 2030 (NDP) places the focus on a model of "economic inclusion" which entails broadening economic opportunities by, inter alia, creating "fertile conditions for entrepreneurship and career mobility" and envisions a decline in production cost to stimulate local production and local procurement. See National Planning Commission Department: The Presidency *National Development Plan 2030 Our Future - Make It Work* (2012) 35, 122 and 129 (*NDP*)

[71] National Planning Commission *NDP* 35.

[72] National Planning Commission *NDP* 35. The formal social compact refers to a united and equal citizenry with a robust leadership and implies that progress should be to the benefit of all. In other words, the model for progress assumed by the National Development Plan favours a zero-cost approach which will entail certain "trade-offs" in pursuit of growth.

[73] National Planning Commission *NDP* 35.

In this context, the "creative and cultural industries"[74] are considered substantial contributors to progress. It is made explicit that "value inculcation",[75] "incentivising commercial distribution networks[76] and "supporting income-smoothing"[77] are considered effective measures to promote the "arts, culture, economy and society."[78]

However, reverse engineering "isn't as popular in the software industry as one would expect […] primarily because software is so complex that in many cases reverse engineering for competitive purposes is thought to be such a complex process that it just doesn't make sense financially."[79] Instead, as the author points out, reverse engineering of software is most common for the purpose of security-based research to identify vulnerabilities[80] or to create programs that rely on tried-and-tested programming techniques, such as, the organisation and structure of commands.[81]

This fact is important in the South African context, which seeks to stimulate the software industry but must do so in a manner that would not increase piracy or imperil the commercial and economic significance of copyright protection. Thus, a local approach to software must maintain sound protection but provide a means for reverse engineering that is mindful of the reality that this process is necessary to create different or unrelated programs that were not designed to be interoperable with the decompiled program, or deliver output that is not in the form of a computer program.

The above approach to progress in the creative industries are to be read with the NDP's approach to an active citizenry, which requires that individuals "work together

[74] 36.

[75] 36.

[76] 36.

[77] 36.

[78] 36.

[79] Eilam *Reversing Secrets of Reverse Engineering* 4.

[80] 5 to 8.

[81] Eilam *Reversing Secrets of Reverse Engineering* 9. The author identifies a difference in this respect between interoperability and analysis of the "robustness" of the program.

with others in the community to advance development".[82] Therefore, it is clear that the virtuous circle model, when read within the developmental context of South Africa, is consonant with a balanced approach to copyright in computer programs, which emphasises the need for a method of input benefit-sharing between creators. In other words, in order to promote progress, development in software must occur within a legislative regime which actively stimulates cumulative development.

This, it is submitted, may be achieved by a re-balancing of the private/public interests to the extent that fair dealing in computer programs is not the only, or the primary, means by which the second and subsequent generations of programs are created. Instead, it is proposed that cumulative innovation relies to a greater extent on the ability of programmers to gain access to existing work and that this fact must be addressed, otherwise it maintains an imbalance that is not aligned with the developmental approach of South Africa.

The model for progress described above adopts a "diversified dynamic"[83] economic structure which, inter alia, recognises that "accelerated technological redundancy and reduced product lifecycles create opportunities for new industrial firms to enter new product segments."[84] In the context of cumulative innovation in computer programs, this sentiment has long been established as a driver for progress. As early as 1989, the US courts recognised "that reverse engineering is 'an essential part of innovation', which 'could lead to significant advances in technology', and that 'the competitive reality of reverse engineering may act as a spur to the inventor'".[85]

[82] National Planning Commission *NDP* 37. In fact, the NDP makes it clear that "the refrain, 'sit back and the state will deliver' must be challenged – it is neither realistic nor is it in keeping with South Africa's system of government."

[83] National Planning Commission *NDP* 122.

[84] 131.

[85] *Bonito Boats Inc v Thunder Craft Boats Inc* (1989) 141 US 489 as cited in N Shemtov *Beyond the Code: Protection of Non-Textual Features of Software* (2017) 70.

In this context, a peculiarity about South Africa's historical relationship with the Berne Convention[86] reveals that, for a period of time, the legislature sought to exploit the flexibilities of Berne in favour of national interests. When South Africa deposited the instrument of accession to the Paris Act (1971) of Berne on 23 December 1974, it contained a "declaration provided for in Article 28(1)(b) of the said Act to the effect that this accession shall not apply to Articles 1 to 21 and the Appendix."[87] Thus, for the period between 24 March 1975 and 1 January 1995,[88] South Africa was only bound by articles 22 to 38, except article 33(1), of the Berne Convention.[89] Since these articles are administrative provisions, the severely limited substantive copyright provisions applicable to South Africa during this period were contained in the Brussels Act (1948).[90]

This meant, inter alia, that the legislature was free to introduce provisions in national copyright law which did not meet the three-step test[91] for a fair and justifiable limitation on the right of reproduction. It also allowed South Africa to regulate licensing, and compulsory licenses for reproduction and translation, in a manner unencumbered by the restrictions[92] imposed by the special provisions regarding developing countries contained in the Appendix to the Paris Act. By exploiting this freedom, South Africa introduced a far-reaching limitation on the right to reproduction in the form of a *reverse*

---

[86] Berne Convention for the Protection of Literary and Artistic Works (Paris Act) as amended on September 28, 1979 (Berne Convention).

[87] WIPO *Berne Notification No. 64 – Berne Convention for the Protection of Literary and Artistic Works Accession of the Republic of South Africa to the Paris Act (1971) (with the exception of Articles 1 to 21 and the Appendix).*

[88] The date on which South Africa became a member of the World Trade Organization.

[89] South Africa has to date not deposited a declaration, in terms of Article 28(1)(c) of the Paris Act, to extend its accession to articles 1 to 21 and the appendix.

[90] South Africa ratified the Brussels text on 20 February 1950 and it entered into force on 1 August 1951. Although South Africa signed the Stockholm Act (1967), which first introduced, inter alia, the three-step test and the wording contained in the Appendix to the Paris Act, it never deposited an instrument of accession.

[91] Article 9(2), introduced by the Stockholm Act (1967).

[92] Article II to V of the Appendix.

engineering exception[93] for certain artistic works and expanded the power to further limit the right to reproduction by regulation.[94] Although these limitations were met with justifiable disquiet,[95] it has set a precedent for the creation of bespoke copyright law that serves the national interest, as opposed to a slavish adherence to foreign and international principles.

Although some of these amendments had a clear interest in defeating the imposition of increasing economic sanctions at the time, it also expressly sought to break "the stranglehold of foreign copyright owners"[96] and thereby stay the tide of litigation against local users. By doing so, copyright law was said to have provided the means for a local industry (in this case in the spare parts sector) to flourish, stimulated the national economy and facilitated wider competition in the market. However, this came at a cost to local authors, which remains unbalanced.[97]

A discussion of the merits of these amendments fall outside the scope of this work. However, it is instructive to note that the current developmental drive to amend copyright law in the national socio-economic interest is not new. Nor was South Africa

---

93 Section 15(3A), introduced by section 2 of the Copyright Amendment Act 66 of 1983 and amended by section 2 of the Copyright Amendment Act 13 of 1988.

94 Section 13, amended by section 8 of the Copyright Amendment Act 56 of 1980.

95 See Dean O H "Copyright Amendment Act, 1983 Protection of industrial and technical works" 1984 *De Rebus* March 114 115; Dean O H "The great copyright controversy" 1988 *De Rebus* July 469 471-2.

96 Dean *De Rebus* (1988) 473.

97 See Dean *De Rebus* (1988) 473 where Prof Dean submits:

> "The real loser as a result of the 1988 amendment has been the South African industrial designer. This amendment was substantially concerned with depriving him and virtually him alone of the protection against having his work reverse engineered."

The argument is made that, in order to reduce the risk of infringement of foreign work, the amendment effectively abrogated the local designer's rights to his work. The inevitable conclusion is that a limitation of the scope of protection, by way of the reverse engineering exception, had a negative effect on the rights of local authors that is not appropriately balanced with the competitive advantage envisioned by the reverse engineering exception.

However, the criticism of section 15(3A) must be understood along with the provisions of the Designs Act 195 of 1993. See Du Bois M "The Appropriate Scope of Property Rights in Registered Designs" 2017 *South African Intellectual Property Law Journal* 5 (1) 34.

alone when it decided to avoid the imposition of articles 1 to 21 and the appendix to the Berne Convention for as long as it could.[98] A large number of other developing nations also chose to find alternative means to achieve the same or similar effect, particularly in relation to the developmental objectives.[99] This work advocates for a similar result, but, as will become clear, it ensures that the result *gives effect* to international law, rather than depart from it.[100]

Similarly, the analysis conducted below in chapter 3, on the rights to adaptation and reproduction of computer programs, must take note of the problems created by the 1980 and 1983 amendments. In particular, it is useful to recognise that any postulated limitation on the exclusive rights has the potential to harm, rather than serve, the interests of local producers.[101]

Furthermore, by operation of article 9(1) of TRIPS, and the single undertaking principle[102] for negotiations regarding WTO membership, South Africa is obliged to comply with the remainder of the Berne Convention (Paris text) since January 1995, despite the fact that it has still not acceded to the full text. This means that any amendment to the Act must meet the minimum standards expressed in the substantive

---

[98] Details to the contracting parties of the Paris Act (1971) of the Berne Convention indicate that, in addition to South Africa, the following countries deposited the same declaration: Bahamas, Bulgaria, Hungary, India, Malta, Netherlands and Thailand.

[99] See Silva A J C "Beyond The Unrealistic Solution For Development Provided By The Appendix Of The Berne Convention On Copyright" 2013 *Journal of the Copyright Society of the USA* 60 (4) 58 1590-1605 for a comprehensive analysis of the divergent measures applied in a wide range of developing nations as an alternative to the appendix.

[100] In chapter 5 below, the inherent flexibilities of the three-step test in the Berne Convention and TRIPS is discussed, and applied, to support the submission that SA copyright law should depart from the approach to decompilation adopted in foreign law.

[101] See the discussion in paragraph 2 1 2 above where it is submitted that the ESC factors must be applied in light of *national* policy and developmental goals. The importance of a focus on national market factors, when determining the nature and scope of exceptions or limitations to copyright protection, is discussed further in chapter 5 below.

[102] The principle is usually articulated as: "Nothing is agreed until everything is agreed." See Wolfe R "The WTO Single Undertaking as Negotiating Technique and Constitutive Metaphor" 2009 *Journal of International Economic Law* 12 (4) 835838.

provisions of the Berne Convention (articles 1 to 21), including the three-step test. For this reason, the argument advanced in this work, regarding a reinterpretation of the scope of justified reverse engineering of computer programs, must balance the developmental objectives outlined herein with the full text of the Berne Convention and subsequent treaties.[103]

Nevertheless, the competitive advantage that reverse engineering or some other exceptions may, or could, afford the local producers of a particular type of work, remains a cornerstone of pro-developmental copyright review in South Africa.[104] Similarly, while an Afro-centric approach to intellectual property law amendments gains justifiable support,[105] the inherent stress in IP law between adequate protection

---

[103] See chapter 5 paragraph 5 2 1 below where the provisions of the Berne Convention are interpreted and aligned with the spirit and objectives of TRIPS and the WIPO Copyright Treaty (WCT) TRT/WCT/001 WO033EN 1996 (WCT).

[104] See for example the comprehensive amendments advanced in Rens A, et al. 2010 *Report on the South African Open Copyright Review*. See also Nicholson D R 2005 *Free Trade Agreements and TRIPS-plus: implications for developing countries in Africa* 3-5 and 11-12 where the author advances many of the same arguments, with a focus on the need for amendment to copyright law in pursuit of supporting educational goals for development. The same arguments are made in Nicholson D R "Intellectual Property: benefit or burden for Africa?" 2006 *International Federation of Library Associations and Institutions* 32 310. See also Nicholson D R "'Fair use' in new Copyright Bill benefits everyone" (2019) *Daily Maverick* <https://www.dailymaverick.co.za/article/2019-09-17-fair-use-in-new-copyright-bill-benefits-everyone/> (accessed September 2019) where the author argues that the model for fair use is preferable for South Africa as a developing nation, although the author relies primarily on the position in developed nations to justify this argument. See further Nwauche E S "A Development Oriented Intellectual Property Regime For Africa" (2005) *11th General Assembly of the Council for the Development of Social Science Research for Africa* (CODESRIA) Maputo Mozambique 6-10 December 2005 at 3 where the author submits:

> "[T]he fact that Africa is a consumer of intellectual property becomes a disadvantage and a major challenge. The unavailability of these goods and services directly impact on Africa's capacity to develop."

This point underpins their subsequent argument for a human-rights based approach to copyright law that must maintain absolute equality between the rights of the copyright owner and the user of protected work and that the three-step test is inimical to so-called "user rights" in the African context. Below, in chapter 5, it is submitted that this view is incorrect and that the three-step test presents a powerful tool for introducing further exceptions and limitations that are primarily aimed at promoting national interests.

[105] See for example Ncube C, et al. 2017 *Intellectual Property Rights and Innovation: Assessing Regional Integration in Africa (ARIA VIII) Working Paper 5* 13; OseiTutu J J "Prioritising Human

and access to knowledge has become more important in determining the course of law review. As a result, the traditional balancing mechanisms[106] are now subject to closer scrutiny, analysed based on values and principles[107] that were not known to, or relevant to, the drafters, and, in many cases, considered to be either ineffective or ill-suited to the demands of a developing nation. In addition, the impact of disruptive technology on traditional competitive and monopolistic business practices,[108] the reduction in costs to enter global markets in ICT-driven services and the rapid increase in digital penetration in Africa, has forced lawmakers to reconsider the impact that a harmonised or homogenised approach to legislation might have on the local economic climate. As some scholars have put it:

> "Since the negotiation of TRIPS in the 1990s, countries at all stages of development, aided by a more engaged civil society, have wised up. They have refused to stand idly by as lopsided IP provisions are packed into the Trojan horse of international trade agreements. Yet, a problem remains. It is clear to negotiators (or at least independent experts) what will not work, but there is little clear vision of the kinds of policies to put in place of the previous century's outdated IP templates."[109]

Development in African Intellectual Property Law" 2016 *World Intellectual Property Organization Journal* 8 (1) 23 26, 28-9; Van Genugten W and Meijknecht A *Harnessing Intellectual Property Rights for Development Objectives - The Double Role of IPRs in the Context of Facilitating MDGs Nos. 1 and 6* (2011).

[106] Such as fair dealing, inherent limitations on the scope and duration of rights and compulsory licensing.

[107] For the most part these relate to a differentiation between the values of pre- and post-colonial eras and the decolonization of intellectual property law. See Ncube C "Decolonising Intellectual Property Law in Pursuit of Africa's Development" 2016 *World Intellectual Property Organization Journal* 8 (1) 34, 36-7. It also, in some cases, refer to a rejection of the perceived neo-colonial effects of harmonisation efforts for developing states such as TRIPS. See Rahmatian A "Neo-Colonial Aspects of Global Intellectual Property Protection" 2009 *Journal of World Intellectual Property* 12 (1) 40. This work is frequently cited to criticize TRIPS and its effect on developing states. However, it must be noted that the author illustrated his argument with reference to China. See also Kongolo T "Historical Evolution of Copyright Legislation in Africa" (2014) *World Intellectual Property Organization Journal* 5 (2) 163 at 174-5 where the author concludes that many African nations have introduced national copyright law after the end of colonization and remained bound to the Berne Convention.

[108] The changes in what intellectual property should, and can, protect are neatly summarized by Forero-Pineda C "The impact of stronger intellectual property rights on science and technology in developing countries" 2006 *Research Policy* 35 808.

[109] Ncube, et al. *Intellectual Property Rights and Innovation* 6-7.

This has created a new tension in IP law review between, on the one hand, the need to empower locals to compete on the global stage by avoiding an isolationist agenda for South Africa and, on the other hand, addressing the high cost of access to knowledge that the law places on South Africa as both a net importer and net consumer of IP.[110] This tension is revealed as the consequence of an unwelcome "'IP socialisation', resulting in ostensibly context-inappropriate IP norms".[111] This has subjected the continued accession and adherence to international IP law, by African countries, to more stringent socio-economic testing.

In this regard, the same observation made by the court in the *Bonito Boats* case,[112] remains part of statutory planning in South Africa. The NDP states that:

> "The policies and institutions that will support the formation of new, dynamic market segments will need to be agile, efficient, dynamic and self-correcting. They must help firms discover **new lines of competitive advantage**."[113]

With regard to intellectual property, government has set, inter alia, the following objectives for legislative reform:

> "14. National IP laws must be appropriate to the level of development and innovation of the country.
>
> 15. An overall transfer of technology strategy should be developed that is aimed at building domestic capacity and skills, enabling stakeholders (industry and academics, but also the

[110] Nwauche *A Development Oriented Intellectual Property Regime for Africa* at 3 provide the following statics with reference to the World Development Indicators Database:

> "Africa's consumptive intellectual property habit is evident in the number of foreign intellectual property rights it protects. In brief, foreign intellectual property rights dwarf African intellectual property. The statistics are indeed grim. For example, the number of foreign patents granted in a number of African countries over the period between 1997 and 2002 is 7564. Out of this number, 7153 are foreign patents while only 411 are local patents. The same is true of trademarks. Out of a total of 86126 trade marks granted in these African States between 1997 and 2001, 8308 are local trademarks while 77818 are foreign trademarks. Other statistics of knowledge goods worthy of copyright protection as well as the infrastructure for their creation are abysmal as well. In 2002, there were 523 telephones per 10000 people in sub Saharan Africa in 2002 and 119 persons had computers per 10000 persons in the same region."

[111] De Beer J, et al. "Evolution of Africa's Intellectual Property Treaty Ratification Landscape" 2018 *The African Journal of Information and Communication* 22 53 55.

[112] See footnote 80 above.

[113] National Planning Commission *NDP* 131 (emphasis added).

> general public) to better absorb knowledge and use it in their particular environment."[114]

To this end, it makes the following remarks intended to guide legal development:

> "It is submitted that an inevitable impact of **stronger protection** and enforcement in terms of the TRIPS Agreement **leads to reducing access to knowledge-related products** in developing countries, thus poor people are exposed to damaging consequences."[115]

In light hereof, it suggests that "South Africa must adopt pro-competitive measures under copyright legislation", and "should allow software to be **adapted to local needs** through copyright legislation that **allows reverse engineering of computer software** programs **consistent** with its international treaty obligations."[116]

These pro-developmental sentiments are set to be implemented with the introduction of the Copyright Amendment Bill 2017.[117] The provisions of the Bill regarding reverse engineering are analysed in detail below in chapter 5.[118]

---

[114] Draft National Policy on Intellectual Property (DNPIP) 4 (original numbering).

[115] DNPIP 29 (emphasis added).

[116] DNPIP 30 (emphasis added).

[117] The Copyright Amendment Bill, 2017 has seen several revisions. At the time of writing, the current version is B13-2017 [as introduced in the National Assembly (proposed section 75) and redrafted by the Portfolio Committee on Trade and Industry *Government Gazette* No. 40121 of 5 July 2016] with revisions published in November 2018. The President returned the Bill to Parliament for reconsideration in June 2020. The reasons for the President's decision are available at: The Presidency "President refers Copyright and Performers' Protection Amendment Bills to Parliament" (2020) *The Presidency of the Republic of South Africa* Available at: www.thepresidency.gov.za/press-statements/president-refers-copyright-and-performers'-protection-amendment-bills-parliament (accessed July 2020). For a discussion of the reasons, see M Palmedo "South Africa's Copyright Amendment Bill Returned to Parliament for Further Consideration" (2020) *Infojustice* Available at: http://infojustice.org/archives /42426 (accessed July 2020) and A Myburg "Opinion: Behind Ramaphosa's rejection of the Copyright Bill" (2020) *Mail & Guardian* Available at: <https://mg.co.za/opinion/2020-07-02-behind-ramaphosas-rejection-of-the-copyright-bill/> (accessed July 2020). For the purpose of this work, it should be noted that the provisions of the Bill which deal with the proposed decompilation exception, the exception applicable to temporary reproduction and the technological protection measures, are not part of the critique levelled at the Bill or the reasons why it was referred back to Parliament. However, the provisions dealing with the introduction of a fair use system, discussed below in chapter 5, and the exceptions for educational use, are some of the reasons why the Bill was returned.

[118] See paragraph 5 3 *et seq.* below.

The memorandum to the Bill[119] states that it "seeks to align copyright with the digital era and developments at a multilateral level"[120] by, inter alia, addressing the "power imbalance"[121] between rights holders and those who benefit from the commercial exploitation of work. It considers the Bill to be consonant with the NDP, the "approach of various Government Departments to IP matters"[122] and strategically aligned with several international mechanisms including the WCT[123] "for purposes of ensuring effective governance, social protection, employment creation and reduction of inequalities."[124]

This is consonant with a view on the copyright protection of creative and utilitarian works in developing countries. The concern is that the incorporation of harmonised copyright protection, or intellectual property rights in general, into regional and international trade-related negotiations no longer permit the full exploitation of the flexibilities[125] envisioned by the Berne Convention, which were meant to enshrine the ability of member states to apply copyright within its own context and in pursuit of its own developmental agenda,[126] in favour of a developed-nations perspective that largely favours a strict protectionist regime.

---

[119] Memorandum on the objects of the Copyright Amendment Bill, *Government Gazette* No. 40121 of 5 July 2016.

[120] Memorandum on the Copyright Amendment Bill 2017 1.1 at 59.

[121] Which is said to be observed in the music industry only.

[122] Memorandum on the Copyright Amendment Bill 2017 1.2 at 59.

[123] 1.2 at 59.

[124] 1.2 at 59.

[125] Silva *Journal of the Copyright Society of the USA* at 584-9 discuss the following flexibilities in the Berne Convention: the power to make exceptions and impose limitations (although this power is limited by the three-step test and which is itself an obstacle to development); the early lapsing of protection for certain works (not protected against retaliation); the compulsory licensing system in the appendix, wider translation rights and the right to make disregard protection in works that were not made available in the language of the developing state. The Berne Convention also places a limited prohibition on the retaliation between members who implement some of these flexibilities.

[126] See also RL Okediji "Development in the Information Age - Issues in the Regulation of Intellectual Property Rights, Computer Software and Electronic Commerce" (2004) *Intellectual Property Rights and Sustainable Development (UNCTAD-ICTSD Project on IPRs and Sustainable Development)* 6 v.

The objections to restrictive copyright law, and the resistance to international law, raised above, are not baseless. There is evidence which suggests that developed nations continue to dictate the direction of multilateral negotiations in law and policy on global trade issues, while the meaningful engagement by developing states is obstructed.[127]

The proof hereof is found in the prevailing decision-making process of the WTO, which is described as one of *passive* consensus:

> "The practice of consensus in the GATT/WTO has always been taken to mean that no party objects rather than that all parties must agree. This abstention option would often be applied by default, because a party was absent from a meeting or generally non-participatory, although in some cases parties might abstain while expressing a measure of disagreement."[128]

This has created a "a hidden system of weighted voting as the reality is that larger countries find it easier to influence voting outcomes than smaller ones."[129] In addition, the veto-right that a consensus system implies, further serves the interests of developed nations because it is "more costly for smaller countries to challenge an outcome popular with large countries than vice-versa."[130] This has brought about a "decision-making equilibrium over the years that responded to underlying power relationships"[131] instead of reaching a fair and equitable decision.

The alternative decision-making process which has been applied by WTO members during some negotiations, namely, a "critical mass"[132] structure based on the market

[127] The discussion here deals with the consensus process, which is a consequence of the single undertaking concept for multilateral negotiations. See Wolfe R "The WTO Single Undertaking as Negotiating Technique and Constitutive Metaphor" (2009) 12 *Journal of International Economic Law* 835 839-840 for an example of how this process has been abused at the cost of developing nations.

[128] Low P "WTO Decision-Making for the Future" 2011 *World Trade Organization Economic Research and Statistics Division Staff Working Paper ERSD-2011-05* 5.

[129] Wolfe *Journal of International Economic Law* 5.

[130] 5.

[131] Low *WTO Economic Research Paper* 5.

[132] Low *WTO Economic Research Paper* 6; Wolfe *Journal of International Economic Law* 849.

dominance of nations and the pro-active exclusion of minor role-players, or so-called free-riding nations, is manifestly in favour of nations that seek to trade on equal footing. Furthermore, this model puts developing nations at an even greater disadvantage insofar as it is “a way of blunting the demand for regional fixes to issues that are best addressed globally.”[133] It also, arguably, places greater emphasis on the “politics of mercantilist bargaining”[134] in favour of those with more to sell and greater purchasing power. This shifts the likely outcome further away from the interests of developing nations, as a consequence of the diffuse reciprocity ideal.[135] In other words, a developing nation is seldom, if ever, in a position to trade like-for-like in volume or value. Consequently, it may be persuaded to pay in the form of submission to law or policy that will advance the interest of more dominant parties in the long run.

Considering “the confluence of IP policy with trade policy,”[136] this creates a risk that the “artificial scarcity”[137] created by harmonised copyright law will not only fail at addressing the needs of South Africa but also directly harm socio-economic development in Africa. In some cases, this has led to the insistence, by South Africa, that IP rights should be excluded from certain trade negotiations.[138]

Therefore, it is submitted that the mismatch between the interests of developed and developing nations widens the digital divide[139] to the extent that “rather than facilitate prospects for diffusion and access to works, the copyright regime has been co-opted to consolidate social gains associated with new technologies and to transform these gains into economic opportunities for owners.”[140]

---

[133] Low *WTO Economic Research Paper* 12.

[134] Wolfe *Journal of International Economic Law* 848.

[135] 849.

[136] Ncube, et al. *Working Paper 5* 54.

[137] Silva *Journal of the Copyright Society of the USA* 584.

[138] Gregory S 2008 *Intellectual Property Rights and South Africa’s Innovation Future* South African Institute of International Affairs 7.

[139] Silva *Journal of the Copyright Society of the USA* 607.

[140] Okediji *Intellectual Property Rights and Sustainable Development* 2.

During negotiations on the WCT and WPPT, the South African delegation made this point clear when it stated that the provisions of the treaty are welcomed where it affords "needed opportunities **for users** in the developing countries, who were often deprived of academic and scientific resources and materials"[141] to exploit works in a wider manner. The SA delegation also welcomed the opportunity to make "amendments to its laws and to deal with the issues on the digital agenda"[142] in a manner which may be different to those of developed nations.

The point is best summarised as follows:

> "The argument is that development interests require an effective system of protection, balanced by robust limitations to **encourage competition and socially beneficial uses**."[143]

If the copyright system fails to do so, as it does in the case of computer programs, the fact must be addressed that it is "highly prejudicial for developing countries"[144] to continue to seek apparent compliance with international trends disguised as obligations, or **borrow from foreign law** in pursuit of keeping pace with technological developments.

For these reasons, this work does not find sufficient reason to augment South African law in light of the trends in foreign law which limit the scope of permissible decompilation. However, this work does not agree that national law must necessarily reduce the level of protection afforded to work in order to stimulate local production. On the contrary, insofar as reverse engineering is concerned, it is shown that by complying with international law and leveraging its flexibilities, local production can be facilitated without detracting from the rights of owners in any way.

It will be shown below that South Africa has failed, insofar as computer programs are concerned, to address the socio-economic impact of copyright and does not adequately heed the model for progress described above. In order to address the

---

[141] WIPO *Records of the Diplomatic Conference on Certain Copyright and Neighboring Rights Questions: Summary Minutes of Main Committee I* (1999) Geneva 611 [395] (emphasis added).

[142] WIPO *Records of the Diplomatic Conference* 611 [395].

[143] Okediji *Intellectual Property Rights and Sustainable Development* 4 (emphasis added).

[144] 3.

mismatch between the purpose of copyright outlined above and the, overly restrictive, copyright protection in computer programs, the current state of copyright in computer programs must first be reviewed, with a focus on selected issues.

2 2 Computer programs in copyright law

2 2 1 The *sui generis* classification

The Copyright Amendment Act of 1992[145] introduced copyright protection for computer programs to the Copyright Act as a *sui generis* type of work. Prior to this, for a period of about 10 years, computer programs qualified for protection as a species of literary work, following the *Northern Office*[146] decision. The reasons why the 1992 Amendment Act elected to depart from the status quo are said to be unclear and has, consequently, been the subject of criticism.[147] Ostensibly, the legislator relied on the early opinion of WIPO[148] which suggested that computer programs may, at the discretion of member states, be protected as a species of literary work or a *sui generis* type of work,[149] despite the fact that the "overwhelming majority of the participants [at the session of the committee of experts and subsequent regional meetings] agreed

---

[145] Copyright Amendment Act 125 of 1992. See footnote 40 above.

[146] *Northern Office Micro Computers (Pty) Ltd v Rosenstein* 1981(4) SA 123 (C) (*Northern Office)*. This decision was confirmed, on assumption of its accuracy, in *Econostat (Pty) Ltd v Lambrecht and Another* 89 JOC (W) at 102 where the court held that "for purposes of this judgment I will accept the correctness of [the Northern Office] judgment although the last word has by no means been spoken on the matter of copyright relating to computers."

[147] For a discussion of the justification for the 1992 Amendment Act and contemporary critique see Jooste C and Karjiker S "Intellectual Property Law in the Digital Environment (EIP Law)" in Dean O H and Dyer A (eds.) *Introduction to Intellectual Property Law* (2014) 393-6. See also Dean O H "Protection of Computer Programs by Copyright in South Africa" (1995) 1 *Stellenbosch Law Review* 86; Pistorius T and Visser C "The Copyright Amendment Act 125 of 1992 and Computer Programs: A Preliminary Overview" (1992) 4 *South African Mercantile Law Journal* 346; Tong L "Copyright Protection for Computer Programs in South Africa: Aspects of *Sui generis* Categorization" (2009) 12 *Journal of World Intellectual Property* 266 269-270.

[148] As expressed, but not canvassed, in WIPO "Committee of Experts on Model Provisions for Legislation in the Field of Copyright - Draft Model Law on Copyright (Preparatory Document) CE/MPC/III/2" (1990) 9 *Copyright* 241 section 3(1).

[149] WIPO "Committee of Experts on Model Provisions for Legislation in the Field of Copyright - Report CE/MPC/III/3" (1990) 9 *Copyright* 282 para 144-8.

that computer programs should be included in the non-exhaustive list of literary and artistic works."[150]

2 2 2 The model for *sui generis* classification

The 1992 Amendment Act, insofar as computer programs are concerned, was expressly "inspired by the model law on protection of computer programs proposed by the World Intellectual Property Organisation (WIPO)".[151] The memorandum does not refer to the model law by number or title which, considering the date of promulgation of the 1992 Amendment Act, gave cause for speculation[152] whether the legislator was referring to the 1978 model provision[153] or the 1990 (draft) model provisions.[154] However, some point out that the memorandum intended to refer to the 1978 provisions.[155] Thus, by implication, the 1992 Amendment Act is said to have introduced *sui generis* protection without due regard for the international consensus expressed by the 1990 model provisions and, instead, elected to follow an approach advocated by a minority of the international community in the 1978 provisions, which is inconsistent with the Berne Convention.

Clearly the inspiration for the current classification of computer programs in South African copyright law must be examined in order to determine the appropriate context in which it should be read and applied.

It is submitted that, first, there is sufficient reason to accept, as explained below, that the 1992 Amendment Act was predicated on the 1978 model provisions but also considered the 1990 model provisions and, second, the position of the Amendment Act was, and still is, consistent with the Berne Convention and TRIPS.

---

[150] WIPO (1990) 9 *Copyright* 282 144. It must be noted that reference is made here to the members' opinion in relation to the 1990 model provisions and not the 1978 model provisions.

[151] Memorandum on The Copyright Amendment Bill 1992 61.

[152] Tong (2009) *Journal of World Intellectual Property* 268.

[153] WIPO "Model Provisions on the Protection of Computer Software" (1978) *Copyright* 6.

[154] WIPO (1990) 9 *Copyright* 282.

[155] Pistorius et al. (1992) *SA Merc LJ* 348; Pistorius T "The copyright protection of computer programs: literary works shunned by the proposed bill" (1991) 1991 *De Rebus* 833-4.

2 2 2 1 A blended approach to the model provisions

In the first premise, the memorandum to the 1992 Amendment Act refers to the "**model** law on protection of **computer programs**."[156] The 1978 model provisions are entitled *Model Provisions on the Protection of Computer Software*, while the 1990 model provisions are entitled *Draft Model Provisions for Legislation in the Field of Copyright.* Clearly the wording of the 1992 Amendment Act corresponds more closely to the title of the 1978 model provisions on both of the most important aspects, namely the specific subject matter (computer software rather than copyright in general) and the status of the provisions (model provisions rather than draft model provisions).

Therefore, the 1978 model provisions were, ostensibly, the basis upon which the legislature elected to adopt protection for computer programs as a distinct category of work, opting to follow the suggestions of WIPO in relation to computer programs specifically rather than the broader sentiments expressed in relation to copyright in general which made only passing reference to computer programs.

It is noteworthy that South Africa was not directly involved in the drafting or negotiation processes of either the 1978 or the 1990 provisions, insofar as it was neither represented on the Committee of Experts (consisting of the Advisory Group of Governmental Experts[157] and the Advisory Group of Non-Governmental Experts[158]) for the 1978 provisions, nor the Committee of Experts[159] which drafted the 1990

---

156 Memorandum on the Copyright Amendment Bill 1992 (the 1992 Amendment Act memorandum) at 61 (emphasis added).

157 The countries represented were: Brazil, Canada, Congo, France, Federal Republic of Germany, Soviet Union, Spain, United Kingdom and United States. For the full list of participants see WIPO "Advisory Group of Governmental Experts on the Protection of Computer Programs (Geneva, March 8 to 12,1971)" (1971) 7 *Copyright* 35 40.

158 The governments of Australia, Brazil, Canada, Japan, Soviet Union, United Kingdom and United States were represented on the panel. In addition, several non-governmental organizations nominated members to the advisory group. South Africa is not among these. For a full list see WIPO "Advisory Group of Non-Governmental Experts on the Protection of Computer Programs (Second Session) - Legal Protection of Computer Programs" AGCP/NGO/II/4 (1975) Geneva Annex 2 page 1.

159 The governments of Algeria, Angola, Argentina, Australia, Austria, Bangladesh, Brazil, Cameroon, Canada, Chile, China, Czechoslovakia, Denmark, Egypt, El Salvador, Finland, France, German Democratic Republic, Federal Republic of Germany, Ghana, Greece, Guinea, Honduras, Hungary,

provisions. The likelihood exists that South Africa was represented indirectly as a member of one of the international bodies represented on one of the above committees, but this is a remote possibility.[160] It is thus not surprising that the memorandum does not refer to the 1978 model provisions directly or by its correct title – the memorandum did not import the 1978 model provisions but used it, read together with the 1990 provisions, to determine the manner in which it will codify the copyright protection of computer programs.

Thus, it is an oversimplification to regard the memorandum as indicative of a legislative step made in ignorance of the international consensus. It is, in fact, more likely a deliberate move to provide legal certainty at a time when the debate about the proper basis for copyright in computer programs prevailed. This conclusion is based on two observations.

First, from the analysis below, it is clear that the 1978 model provisions were, and still are, significantly more expansive, sophisticated, detailed and technologically nuanced in relation to computer programs than the 1990 model provisions.[161] The 1990 model provisions addressed several other issues in the field of copyright, while the 1978

India, Israel, Italy, Japan, Lebanon, Mexico, Morocco, Netherlands, Norway, Poland, Portugal, Republic of Korea, Saudi Arabia, Soviet Union, Spain, Sweden, Switzerland, Thailand, Tunisia, Turkey, United Kingdom, United Republic of Tanzania, United States of America, Uruguay and Yugoslavia were represented. For a full list see WIPO "Committee of Experts on Model Provisions for Legislation in the Field of Copyright (First Session)" (1989) 25 *Copyright* 146 146.

South Africa also failed to respond to the call for comments on the provisions. See WIPO "Committee of Experts on the Legal Protection of Computer Software (Second Session) - Analysis of the Results of the Survey Concerning the Desirability and Feasibility of a Treaty for the Protection of Computer Software and/or Other Possible Measures in the Field of the Protection or Deposit of Computer Software" LPCS/II/2 (1983) Geneva 2 fn1.

[160] The majority of intergovernmental and international non-Governmental organizations listed are European-based, with the only notable exception being the Organization of African Unity (OAU), which is listed on the Committee of Experts in relation to the 1990 provisions. South Africa was not a member of the OAU at the time and was only admitted in 1994.

[161] The 1978 model provisions are entirely devoted to computer programs and address all of the traditional copyright principles in relation to software directly, along with several pages of explanatory commentary on each of the sections. By contrast, the 1990 model provisions refer to computer programs only twice and does not go beyond a mere indication of the classification of this type of work.

model provisions dealt with computer programs only. The 1990 model provisions had a clear intention, namely, to address legislative development in the field of copyright. By contrast, the 1978 model provisions considered computer programs afresh in light of the peculiarities of the type of work and the available alternative intellectual property regimes. Therefore, the 1978 model provisions presented a more considered approach to the legal protection of computer programs, based on a review of the full scope of intellectual property rights, upon which South Africa could rely.

Second, it was the intention of the International Bureau of WIPO that the 1978 model provisions would form the basis of an international treaty for the protection of computer programs, including a global depository system. For this reason, the 1978 model provisions specifically adopted a definition of originality that would "promote a desirable uniformity of protection"[162] in various countries by being read "within the meaning of the copyright law of the country concerned".[163] Extensive work[164] was done toward furthering the international harmonisation of protection of computer programs after the publication of the 1978 model provisions, including the Draft Treaty on the Protection of Computer Software,[165] which was discussed, based on the questionnaire responses of several nations, in 1983 at the second session of the Committee of Experts on the Legal Protection of Computer Software.[166] The draft treaty elicited extensive discussion, but was eventually abandoned in favour of a system of national protection.[167]

---

[162] WIPO (1978) *Copyright* 16.

[163] WIPO (1978) *Copyright* 16.

[164] See WIPO "Expert Group on the Legal Protection of Computer Software (First Session) - Measures to Enhance International Cooperation in the Field of Legal Protection of Computer Software" LPCS/1/2 (1979) Geneva.

[165] WIPO "Committee of Experts on the Legal Protection of Computer Software (Second Session) - Draft Treaty for the Protection of Computer Software" LPCS/II/3 (1983) Geneva.

[166] WIPO *LPCS/II/2* page 7 at 14(f).

[167] The demise of the treaty was largely due to the work of a joint research project between WIPO and UNESCO on the copyright protection of computer programs. See Miyashita T "International Protection of Computer Software" (1991) 11 *The John Marshall Journal of Information Technology and Privacy Law* 42 52-3 for a discussion of the joint research which found that "there were no immediate and effective means with which to achieve a harmonized international protection system".

Thus, considering that the 1978 model provisions "essentially adopted a copyright approach,"[168] it presented a comprehensive analysis on how this should be achieved and suggested that it will be the basis of legislative amendments worldwide in the future. In this light, the South African legislature's decision to adopt an approach aligned with the 1978 model provisions is understandable. Consequently, the 1978 model provisions have an important interpretive role to play, for the purpose of this study, in order to re-consider the purpose of software copyright as a driver of progress. Furthermore, the 1990 model provisions were not intended to contradict or subsume the 1978 model provisions. In fact, the 1990 model provisions expressly[169] "utilized"[170] other documents, including those related to "the protection of computer programs"[171] and "the means of combating piracy."[172] The deliberations on the 1990 provisions make it clear that the 1978 model provisions "identified the most important provisions to be applied,"[173] namely, the scope and definition of the type of work,[174] the level of originality,[175] authorship,[176] and term of protection.[177] In all other respects, the 1978 model provisions do not depart from the contemporary standard or default position in copyright law.

As a result, the 1990 model provisions proceeded to address only one question, namely, whether or not the 1978 model provisions suggest that computer programs be subject to *sui generis* legal protection or copyright protection (either as a *sui generis* type of work or a literary work). Early in its deliberations on the 1990 model provisions, the committee of experts found that "just the latter solution has been accepted and

[168] WIPO (1978) *Copyright* 16.

[169] WIPO "Committee of Experts on Model Provisions for Legislation in the Field of Copyright First Session (Geneva, February 20 to March 3, 1989)" (1989) 25 *Copyright* 146 149.

[170] WIPO (1989) *Copyright* 148.

[171] 148.

[172] 148.

[173] WIPO (1989) *Copyright* 149.

[174] 1978 Model Provisions section 1(i)-(iv).

[175] Section 3.

[176] Section 2(1).

[177] Section 7.

has become nearly exclusive"[178] and that a "clear and unmistakable trend" has been identified in favour of protecting computer programs in terms of copyright law.

Subsequently, deliberations on the 1990 provisions deal with copyright in computer programs *per se* as a *fait accompli,* and address only one question, namely, the type of work. On this point, the committee held "no in-depth discussion [about] which of the alternatives was more appropriate,"[179] and invited further comments on the question. As a result, the original draft of the 1990 provisions prevailed which included computer programs in the non-exhaustive list of works as either a species of literary work[180] or a sui generis type of work.[181] It is important to note that the 1990 model provisions did not, at any point, determine that computer programs should be classified as either a literary work *or* a separate category. In fact, it invited commentary on the point of selecting either option or *maintaining both alternatives*.[182]

2 2 2 1 2 The value of the model provisions as interpretive tools

The conclusion one may draw from this is that South Africa, when deciding on the legislative approach to computer programs, determined that, with reference to the 1990 model provisions, copyright protection is considered to be the most appropriate vehicle. Thereafter, when it came to describing the manner in which copyright law shall address computer programs, it relied on the leading authority in this respect, namely, the 1978 model provisions which expressed, at the time, the most authoritative view on the nature of computer programs and its legal protection.

2 2 2 1 3 The motivations for introducing sui generis classification

Regarding the election to list computer programs as a *sui generis* type of work, it was justified for several reasons.

---

[178] WIPO (1989) *Copyright* 149-150.

[179] WIPO (1990) *Copyright* 261 at 146.

[180] 1978 Model Provisions section 3(1)(i).

[181] Section 3(1)(xii).

[182] WIPO (1990) *Copyright* 261 at 147.

2 2 2 1 3 1 Adherence to contemporary international precedent

First, there is no contemporary indication that the international community favoured copyright protection for computer programs as a species of literary work over *sui generis* classification.[183]

In fact, the leading model law at the time, that of 1978, gave every indication that, in the event that *sui generis* protection for computer programs should not become the norm, copyright protection of computer programs should be in a *sui generis* manner,[184] and the 1990 model provisions did not change this position. Therefore, South Africa's decision to adopt copyright protection as a *sui generis* type of work is aligned with both the 1978 and 1990 model provisions.

2 2 2 1 3 2 Expanding on local precedent

Second, prior to the 1992 Amendment Act, South Africa already protected computer programs in terms of copyright law.[185] A departure from this system, to introduce *sui generis* protection, would have been contrary to leading local precedent and would require, at least, the introduction of a new statute rather than a mere amendment bill.

Instead, the 1992 Amendment Act, by codifying *copyright* protection for computer programs, addressed the need for law reform on three fronts, namely providing legal certainty about the status of computer programs in law, clarifying the position developed in case law and incorporating into national law the most sophisticated international mechanism for protection without departing in any significant manner from local jurisprudence.

2 2 2 1 3 3 Responding to critique

Third, it must also be noted that the South African position prior to the 1992 Amendment Act was subject to criticism. Some commentators focused on the fact that

---

[183] In fact, as pointed out above, consultations on the 1990 model provisions never expressed a clear preference for either classification and invited further commentary on this point while retaining the dual classification.

[184] The model provisions expressly adopted a copyright-based approach which "takes account of [its] subject matter's affinity with copyright protection". See WIPO (1978) *Copyright* 8.

[185] See the discussion in paragraph 2 2 above.

protection as a literary work was unsound[186] because "the more the computer program is transformed in order to read it into the computer, the more it attains a technical character,"[187] which makes it ill-suited for protection as a set of humanly legible instructions.

Others took a more comprehensive approach, based on the manner in which computer programs are created and commercialised, and found, inter alia, that classification as a literary work creates an authorship/ownership divide that does not heed the needs of programmers who often work in teams.[188] It was also submitted that, despite the fact that literary works need not express any literary quality, computer programs are not, in all cases, an expression akin to writing.[189]

Furthermore, South African case law after the *Northern Office* decision[190] expressed difficulty in protecting computer programs as a species of literary work for a number of reasons. These problems, which contributed to the legislator's decision not to include computer programs as literary works, are summarised by Tong as follows:
(1) "difficulty in accommodating object code within the category of literary works";
(2) "concerns about the relationship between computer programs and specifications";
(3) "the scope of protection given to computer programs as a result of the categorization as literary works"; and

---

[186] See Staines A "Copyright in Computer Programs: A Hollow Victory" (1983) 46 *Modern Law Review* 231 232-3 and Visser C "A comparative survey of aspects of the subsistence of copyright in computer software" (1984) 17 *Comparative and International Law Journal of Southern Africa* 32 37-8 and 49. See also Tong (2009) *Journal of World Intellectual Property* 268 and the sources mentioned therein.

[187] Visser *Comparative and International Law Journal of Southern Africa* 37. It must be noted that this statement is made by Visser in order to argue that no traditional system of intellectual property protection is suitable, including patent and copyright protection.

[188] Horowitz G B "The Copyright Act of 1978 as it affects computer programs and data" (1983) 100 *SALJ* 301 305.

[189] Horowitz (1983) *SALJ* 302-3. See also *Econostat (Pty) Ltd v Lambrecht and Another* 89 JOC (W) at 102 and the scholarly opinions cited by the court in this respect.

[190] In particular the decisions in *Econostat (Pty) Ltd v Lambrecht and Another, Apple Computer v Rosy t/a SA Commodity Brokers (Pty) Ltd and Another* 1984 JOC (13) 134 (D), discussed further below, and *Pastel Software (Pty) Ltd v Pink Software (Pty) Ltd and Another* 1991 JOC (13) 398.

(4) “the issue of authorship which held potential pitfalls, particularly because computer software is usually developed by more than one person.”[191]

This created an opportunity for the legislator to do more than merely codify common law, and also address those aspects of computer programs that caused uncertainty in copyright law. It is thus not surprising that the memorandum to the 1992 Amendment Act referred to “certain peculiarities”[192] of computer programs which necessitated that this type of work should be protected independently of literary works.

2 2 2 1 3 4 Preserving international harmonisation

Fourth, the deliberations on the 1990 model provisions expressed a reservation regarding the classification of computer programs as literary works, dreading a dilution of copyright in traditional works if computer programs are included as a type of literary work.[193]

In this respect, commentary expressed doubt whether or not the principle of national treatment, prescribed by the Berne Convention and TRIPS,[194] would be applicable to computer programs as a species of literary works. The committee feared that some nations would find a reason to depart from the principle of national treatment as a result of the inclusion of computer programs because it can no longer be obliged to apply it universally to the works of all foreigners.[195] Considering the lack of technical sophistication of the 1990 model provisions, insofar as computer programs are concerned, this fear was not unfounded. It did not provide clarity on what a computer program is for purposes of copyright law or how it would be tested against the requirements for vesting copyright protection.

Consequently, the risk existed that some member states of the Berne convention would be forced to depart from the minimum level of protection, prescribed by the

---

[191] Tong (2009) *Journal of World Intellectual Property* 269.

[192] Memorandum on The Copyright Amendment Bill 1992 61.

[193] See WIPO (1990) *Copyright* 148.

[194] Article 5 of the Berne Convention and Article 3 of TRIPS.

[195] See WIPO (1990) *Copyright* 148.

Berne Convention, or refuse to apply the principle of national treatment, merely because the member state had elected not to include computer programs as a species of literary work.

For example, should a national of country A, where computer programs are protected as literary works, seek to enforce his copyright in a computer program in country B, country B would be obliged to recognise that right to the extent that its national *copyright* law allows. However, if country B does not recognise computer programs as a copyrightable type of work, for whatever reason, it is left with one of two choices: either protect the work as a species of literary work under pressure, by operation of the Berne Convention, or refuse to apply the principle of national treatment in relation to all literary works.

In other words, where a member state elected not to include computer programs as literary work, that state would be empowered to disregard the principle of national treatment in relation to all literary works. The only alternative interpretation is that country B protect the computer program of a foreign author as a literary work, despite the fact that country B's legislator has elected not to do so. This would be contrary to the principle of self-determination and, in the absence of a deeming provision in Berne obliging member states to recognise computer programs, give country B sufficient cause to disregard the principle of national treatment or depart from the minima under Berne. Consequently, South Africa's decision to introduce computer programs as a *sui generis* type of copyrightable work may be understood as an attempt to avoid departing from the principles of the Berne Convention.

Ultimately, the 1990 model provisions would not reach the same stage of development as the 1978 model provisions and the Berne Convention was amended to list computer programs as a species of literary work. The effect of the classification of computer programs in the Berne Convention is discussed in more detail below.[196]

[196] See paragraph 2 2 2 2. The wording of the relevant provisions in the Berne Convention is discussed in more detail in chapter 5.

2 2 2 1 4 The South African position in the international context

The above discussion makes it clear that, when South Africa elected to introduce computer programs as *a sui generis* type of work to the Copyright Act, it did not do so based solely on the 1978 model provisions. In fact, South Africa's approach clearly considered the influence of the 1990 model provisions when it elected to introduce copyright protection, as opposed to *sui generis* protection or protection as a literary work, and provided legal certainty in relation to the protection of computer programs aligned with leading global consensus and a sensitivity to the nature of computer programs and South Africa's international obligations.

Furthermore, the legislator's decision to depart from the traditional literary approach in favour of a model inspired by the 1978 model provisions is, when seen in the above context, a deliberate step toward safeguarding copyright in computer programs which anticipated a treaty on the protection of computer programs based on the 1978 model provisions.

An alternative view leads to the same conclusion. The 1978 model law failed to gain sufficient support, not because it proposed a system that was not amenable to members, but because a significant number of states had already made progress on the copyright protection of computer programs and a desire that the work of the committee should instead focus on a study of the various national laws and compliance with existing treaties.[197] Therefore, the 1978 model provisions, insofar as it advocated for a *sui generis* system, was no longer required because a copyright-based approach, which is the basis of the 1978 model provisions, had gained sufficiently wide acceptance.[198]

According to the joint report of the group of experts on computer programs, convened jointly by the Secretariat of UNESCO and the International Bureau of WIPO, a "great number of participants developed arguments in favour of recognizing copyright

[197] See Miyashita (1991) *The John Marshall Journal of Information Technology and Privacy Law* 52.

[198] See for example WIPO "Group of Experts on the Copyright Aspects of the Protection of Computer Software (Geneva, February 25 to March 1, 1985) - Report" (1985) 21 *Copyright* 146.

protection of computer programs."[199] Thus, considering that the international community expressed a clear desire that the protection of computer programs "should be based on an internationally harmonized approach,"[200] South Africa's decision to adopt a copyright approach is not peculiar or out of step with the international community. It does, however, express a belief that South African copyright law, and jurisprudence, would be capable of development to (1) keep pace with the development of computer technology and (2) ensure that the level of copyright protection afforded to computer programs remain, at least, on par with that of literary works.

2 2 2 2 Compliance with international law

In the second premise, it has been shown that, at the time of the 1992 Amendment Act, the 1990 model provisions expressed a concern about the classification of computer programs as literary works in relation to adherence to the Berne Convention, and failed to provide clarity on this point. The 1990 model provisions also retained the possibility of classification as a *sui generis* type of work, while the 1978 model provisions outlined this form of protection in a manner that avoided the possible conflict with the Berne Convention, explained above. It must be noted that the 1990 model provisions did not result in an amendment of the Berne Convention in relation to computer programs.

Copyright protection for computer programs would only become subject to the provisions of the Berne Convention after the TRIPS agreement entered into force. The current text of the Berne Convention does not include computer programs in the illustrative list of literary works,[201] or anywhere else in the text. Computer programs are capable of protection as a species of literary works by operation of article 10(1) of TRIPS, which indirectly include computer programs as a type of literary work in terms of the Berne Convention.

---

[199] WIPO (1985) *Copyright* 147.

[200] 147.

[201] Article 2(1).

This brings into question the status of the South African position post 1992 in relation to its international obligations.[202] It is submitted that the *sui generis* classification is not at odds with the spirit, purport or meaning of TRIPS or the Berne Convention in any respect, for the following reasons.

2 2 2 2 1 The meaning of protection as literary works

First, article 10(1) states that computer programs "shall be protected as literary works under the Berne Convention." Some have argued that the wording of this provision implies that computer programs shall qualify for copyright protection *to the same extent* as literary works, rather than as a form of literary work.[203]

Thus, provided that the manner in which a member state extends copyright to programs is on par with that of literary works, in nature and scope, rather than manner or categorisation, it would be compliant with TRIPS. It is submitted that this argument is sound, and may be expanded.

There is no indication that article 10(1) was ever intended to be a deeming provision in relation to the classification of works. In fact, the negotiation process on article 10(1)[204] makes it clear that the drafters were only concerned with ensuring that member states not be allowed to deviate from the minima expressed by the Berne Convention insofar as the term of copyright in computer programs are concerned, or be permitted to vest copyright in the constitutive elements *per se* of a computer program, namely, "procedures, methods of operation or mathematical concepts."[205]

---

[202] See Tong *Journal of World Intellectual Property* 271 where the author submits that the effect of the classification of computer programs as sui generis type of work is at odds with TRIPS in relation to certain aspects. See also De Villiers *SALJ* 336 fn 150 where the author submits that sui generis classification does not satisfy the obligation created by section 10 of TRIPS.

[203] See Tong *SALJ* 518.

[204] See Okediji R "Copyright: Computer Programs" in Ricupero R and Ortiz RM (eds.) *Resource Book on TRIPS and Development UNCTAD-ICTSD* (2005) 153-4 for a summary of the negotiation process and the text of the various draft versions of article 10(1).

[205] TRIPS article 9(2).

The intention of article 10(1) was never to determine that computer programs *shall be* literary works for purposes of copyright. Instead, it suggests that member states shall provide for protection of computer programs as literary works, in other words, *to the extent* that the Berne Convention prescribes the scope of protection. This point is supported by the wording of TRIPS itself, which provides that member states are "free to determine the appropriate method of implementing the provisions of this Agreement within their own legal system and practice"[206] and may elect to provide more extensive protection.

The above interpretation of TRIPS is supported by the wording of the WCT which states that "computer programs *are* protected as literary works within the meaning of Article 2 of the Berne Convention."[207] During the negotiation process of this treaty, several nations questioned the use of the word "are" in this article, and suggested that it should be replaced with "shall be", to conform to the wording in TRIPS.[208] This suggestion was not accepted, which makes it clear that the WCT provision is a statement of fact,[209] rather than a deeming provision on the classification of the type of work.

It will be shown below that classification as a *sui generis* type of work has extended greater protection for computer programs and, as a result, cannot be said to be in contravention of TRIPS or the Berne Convention.

2 2 2 2 2 The absence of a deeming provision

Second, TRIPS does not define computer programs, and yet it addresses the peculiar nature of computer programs directly by including reference to both the source and object code of computer programs. If article 10(1) was intended as a deeming

[206] Article 1(1).

[207] WCT Article 4.

[208] See WIPO "Records of the Diplomatic Conference on Certain Copyright and Neighboring Rights Questions: Summary Minutes of Main Committee I" (1999) Geneva 640-2.

[209] See chapter 4 below, paragraphs 4 2 1 and 4 3 1, where it is shown that the classification of computer programs as literary work developed, in the US and UK, from a general assumption that the literary nature of programs is analogous to other written work.

provision, reference to the source or object code would make no sense because both forms of code are *per se* literary in nature.

It is clear that literary works may be conveyed in any number of languages and will qualify for copyright protection regardless of the meaning borne thereby or the language, or alphabet, it is composed of. Similarly, computer programs may be drafted in any number of programming languages and, once compiled into object code, will nevertheless remain a work consisting of literary and/or numerical elements.

Thus, if TRIPS intended that, henceforth, computer programs *shall be* literary works, there would be no need to address object code or source code at all, because all versions or translations of a program will automatically qualify for protection like any other literary work. The fact that article 10(1) is not intended to be read in this manner explains why computer programs are not deemed to be literary works but merely indicated as capable of protection as a species of literary work.

The intention is clearly to provide that, in the event that a member state provides protection for computer programs as literary works, it shall consider both the source code and the object code as eligible expressions in the form of a literary work regardless of the fact that source and/or object code may consist partially or entirely of mathematical formulae. Therefore, TRIPS provides the necessary "flexibility" for member states to "determine the extent of copyright protection in a particular computer program"[210] and stipulates that the extent shall be, at a minimum, akin to that of literary works in terms of the Berne Convention.

Article 10(1) is clearly a strong endorsement for the view that computer programs are suitable subject matter for copyright protection and, as such, settled the underlying debate which surrounded, inter alia, the 1978 and 1990 model provisions. However, it has been shown that TRIPS did not impose an obligation on member states to enact protection for computer programs as literary works. It considered that member states may wish to provide other forms of protection,[211] and suggested that, insofar as

[210] Okediji "Copyright: Computer Programs" 158.

[211] Okediji "Copyright: Computer Programs" 155.

copyright protection is concerned, the established principles in relation to literary works be applied as a benchmark when national law is developed.

This approach has the additional benefits of removing the need for every country to determine, and provide for, the technical requirements of software and implementing copyright protection for computer programs immediately, without the need for legislative amendment. Consequently, all member states could, without more, meet their obligation in terms of article 10(1) of TRIPS, adhere to the principle of national treatment in relation to computer programs and avoid a conflict of national laws. This view is supported by WIPO and UNESCO which, by way of the group of experts, stated that "protection under existing international copyright conventions would promote the production and international circulation of programs without delay, by means of extending the protection granted to national creators of computer programs to nationals of other Contracting States".[212]

There is no reason to suggest that article 10(1) should be interpreted any differently. However, one may ask, if the above construction was intended, why does the wording of article 10(1) not provide that computer programs shall be protected "to the same extent" as literary works or use similar wording. The answer is twofold; any word other than "as" in article 10(1) would create uncertainty about the scope of protection that is proposed by TRIPS and would exclude computer programs from the ambit of literary works.

One may also argue that article 10(1) is a deeming provision regarding the classification of software, despite the above conclusions, and refers to source and object code purely to provide legal certainty about the status of translations. However, at the time when TRIPS was negotiated, this issue was no longer of concern to the international community[213] and had been adequately addressed in a large number of

---

[212] WIPO (1985) *Copyright* 147.

[213] The leading international precedent on the technical nature of computer programs for purposes of copyright protection at the time, namely the 1978 and 1990 model provisions and related negotiation documentation put it beyond doubt that source code and object code is eminently capable of protection as versions of the same work. See for example: WIPO (1978) *Copyright* 14.

jurisdictions.[214] The status of translations in relation to literary works was an established principle and expressly provided for in the Berne convention in relation to the right of public recitation and of communication to the public of a recitation.[215]

Therefore, the reference to source and object code in article 10(1) of TRIPS is not indicative of an intention to stipulate that computer programs are to be protected as literary works. On the contrary, the Brussels text of article 10(1) of TRIPS contained the word "literary" in square brackets and retained the reference to object and source code. Thus, it proposed that computer programs be protected as works under the Berne Convention in either (and both) forms of code, but does not prescribe the type of work. In the final text the brackets are removed. This does not indicate an intention to classify computer programs as literary works. It was a move to avoid the risk that computer programs would be protected as works of applied art,[216] and, consequently, not be eligible for the full term of protection afforded to literary works.

2 2 2 2 3 A retrospective review

Consequently, the election to include the word "literary" in article 10(1) is not a classification of the type of work but an indication of the minimum level of protection that must be afforded to computer programs.

This conclusion is supported by contemporary opinion. During its first session the members of the committee of experts responsible for the 1990 model provisions[217] considered, inter alia, the term of protection that should be afforded to computer programs. The dissenting opinion argued that "the basic provisions of copyright protection"[218] are not suitable because, "the 50-year term of protection after the author's death is unrealistic; computer programs become obsolete within a much

[214] At the time, computer programs qualified for protection in at least 54 countries. See Okediji "Copyright: Computer Programs" 153; Miyashita *The John Marshall Journal of Information Technology and Privacy Law* 52-3.

[215] Berne Convention Article 11ter(2).

[216] Okediji "Copyright: Computer Programs" 271.

[217] WIPO (1989) *Copyright* 148-9.

[218] 148.

shorter period."[219] This view persists to this day[220] and is one of the fundamental points of critique against the copyright protection of computer programs from a socio-economic perspective. It is also the foundation of the abuse of rights doctrine[221] in relation to software, which is discussed further below.[222]

However, the committee of experts dismissed this argument:

> "The alleged problem of the long term of protection is of an academic nature; there are a number of other categories of literary and artistic works which may become obsolete within a much shorter period than 50 years after the author's death which should be considered nothing else but an upper limit. In the overwhelming majority of countries whose legislation or courts have decided on the protection of computer programs, copyright protection has been chosen."[223]

In light of this, the decision to refer to literary works in article 10(1) is proven to be a move to determine the minimum level of protection that member states shall afford to computer programs, regardless of the problems it may create in future. It does not amount to a classification of computer programs as literary works.

Be that as it may, the accompanying memorandum to the 1992 Amendment Act states that:

> "The evolution of computer programs as a sui generis category of work out of literary works is consistent with the development of the law in the past when cinematograph films evolved as a category out of dramatic works and sound recordings evolved as a category out of musical works."[224]

---

[219] 148.

[220] See for example Ncube (2012) *Stell LR* 452 where the author submits that the "software industry is fast-paced and computer programs have a short shelf life accompanied by frequent updates. They also have low development costs. Accordingly, short to medium term protection is appropriate." (original citations omitted).

[221] Shemtov *Beyond the Code* 138-9.

[222] See the discussion in chapter 4 below, in particular the critical analysis of anti-circumvention provisions in paragraph 4 2 4 *et seq*.

[223] WIPO (1989) *Copyright* 149 (emphasis added).

[224] Memorandum on the Copyright Amendment Bill 1992 at 62 (emphasis added).

Therefore, it appears that the South African approach was motivated by a sense of divergent development in human expression and the impression that a homogenous view of literary works could no longer address the reality of computer programs for purposes of copyright protection. This is supported by the 1992 Amendment Act memorandum which states:

> "It is considered that computer programs should be protected by copyright but that it is artificial and not entirely appropriate to treat a computer program as if it were a written text. Many of the provisions of the law relating to the protection of literary works are unsuitable for application to computer programs and conversely there are certain peculiarities of computer programs which are not provided for in treating them as literary works."[225]

However, some South African scholars maintain that TRIPS "*classify* computer programs as literary works"[226] which "raises some concerns about the Copyright Act's compliance with TRIPS."[227] It has been shown above that, although the South African approach was atypical,[228] it is not inconsistent with the text or the intention of TRIPS or the Berne Convention or, by implication, the WCT.[229]

This conclusion is endorsed by WIPO regarding the wording of TRIPS and WCT. Despite the fact that both instruments[230] provide that computer programs shall be (or are)[231] protected as literary works, this "does not exclude that national laws may

[225] 61 (emphasis added).

[226] Ncube (2012) *Stell LR* 441 (emphasis added). See also De Villiers (2006) *SALJ* 326; 336 n150.

[227] Ncube (2012) *Stell LR* 441. It should be noted that the author does qualify her statement, cited here, in a later publication, to the extent that IP rights in developing nations should no longer seek to mirror the international consensus but, instead, seek minimum protection. Minimum protection in this context is understood as minimal protection rather than minima. See Ncube C "Harnessing Intellectual Property for Development: Some Thoughts on an Appropriate Theoretical Framework" (2013) 16 *Potchefstroom Electronic Law Journal* 370 372-3.

[228] Inasmuch as it opted to depart from local case law and international precedent to extend copyright protection to computer programs as a *sui generis* type of work.

[229] The relationship between the provisions of the Berne Convention, TRIPS and the WCT, regarding decompilation, is analysed below in chapter 5.

[230] Article 10(1) of TRIPS and article 4 of the WCT.

[231] The difference in the wording of each instrument is important and discussed in chapter 5.

categorize computer programs as a separate category of works."[232] The proper interpretation of these articles are, thus, explained as follows:

> "The significance of this categorization of computer programs as literary works (writings), depends on other relevant provisions of the respective laws and on the practice adopted in subsequent court decisions. It should be kept in mind that <u>this categorization of computer programs indicates that the level of originality</u> required as a prerequisite for protection should <u>not be different from that required for other writings</u>."[233]

From the above, one may conclude that the 1992 Amendment Act should not be construed to place computer programs on par with literary works. It is neither an obligation to treat computer programs as literary works nor was it intended that computer programs should be treated as analogous to literary works.

On the contrary, the amendment clearly sought a new vehicle for the protection of computer programs, within the flexibility of the copyright regime. The reason for this is clear:

> "The treatment of a computer program as a literary work was […] not entirely satisfactory because the provisions of copyright law relating to literary works have been framed over the years with written texts in mind and the <u>peculiarities of computer programs are such that clothing them in the garb of literary works has been an uncomfortable fit</u>."[234]

This view was not welcomed. Leading South African scholars expressed a reservation about the direction adopted by the 1992 Amendment Act, because "for obvious reasons South Africa should strive to adopt protective measures for computer programs that are in line with those of our important trading partners and in order to meet our obligations in terms of the Berne Convention."[235] It has been shown that the Berne Convention does not impose such an obligation on South Africa.

However, the argument is made that the 1992 Amendment Act introduced a "strained relationship"[236] between copyright law and computer programs. This view considered the peculiarities of computer programs, i.e. the effects of its utilitarian nature, and felt

---

232 WIPO *Intellectual Property Handbook (WIPO Publication No. 489 (E))* (2008) 437 at 7.15.

233 WIPO *Intellectual Property Handbook* 438 at 7.18.

234 Dean OH "Copyright Amendment Act, 1992" (1992) Oct *De Rebus* 755.

235 Pistorius (1991) *De Rebus* 834.

236 Pistorius et al (1992) *SA Merc LJ* 348.

that, despite this, computer programs are best suited to protection as literary works because "literary works of utilitarian nature have always enjoyed copyright protection, for example "how-to" books, histories, maps and other factual matter and the functionality of computer programs should, therefore, be regarded as irrelevant."[237]

It must be observed that, at the time, these statements did not, and could not, reflect on the future of computer programs and the pivotal role it would play in copyright jurisprudence. Instead, these sentiments were justified from a protectionist point of view, in an attempt to safeguard the integrity of the copyright regime and make sense of a development which was, and remains, unfamiliar.

In addition, the critique levelled at the 1992 Amendment Act must be considered in context. It was made at a time when the socio-economic importance of copyright did not feature as prominently as it does today and, in South Africa, was not subject to the same developmental agenda. The critique is, thus, understandable. It is useful to the current study to observe that these sentiments are no longer paramount and have been replaced with a greater emphasis on the impact of copyright on socially beneficial endeavours as opposed to trade relations or international harmonisation.[238]

In light hereof, and with the benefit of hindsight, it is determined that the intention of the legislator was to provide for protection that is unique, capable of addressing the development of technology and the developmental needs of the nation. In addition, the construction of the Berne Convention and TRIPS above make it clear that the South African position is not at odds with international law.

2 2 3 The consequence of a misapplication of the sui generis classification

In light of the above limitation, computer programs are to be afforded *no less* protection than literary works *insofar as the originality test* is concerned. Nothing more is envisioned or should be read into South Africa's obligations. In other words, computer programs may not be disqualified from protection because they do not meet the same

[237] Pistorius (1991) *De Rebus* 834.

[238] See the discussion above regarding the ESC impact factors, and the analysis of the pro-developmental flexibilities in the three-step test, discussed in chapter 5 below.

originality standard as literary works because, for example, they are not capable of conveying a meaning to all humans or to humans in all of their translated versions. The opposite is not true. It was never intended that computer programs should be subject to a higher level of originality, a wider range of restricted acts or be made subject to the same jurisprudence on the idea/expression dichotomy developed to address literary works.

Unfortunately, the development of copyright in computer programs in South Africa did not consider this crucial qualification, namely that computer programs are no longer to be treated as literary works.[239] In fact, the full, but limited, volume of opinion and case law post 1992 never correctly addressed the nature of computer programs as *a sui generis* type of work despite the fact that computer programs are expressly excluded from being a literary work.[240] In all cases, it was merely accepted that computer programs are an eligible type of work and, where necessary, may, or must, be treated as analogous to literary works in relation to all copyright principles. This error allowed the courts to borrow from foreign law to develop local jurisprudence to the extent that case law on computer programs as literary works, in foreign law, carried persuasive value.[241]

Consequently, the amalgamation of computer programs and literary works imported a doctrinal approach to the protection of computer programs into South African copyright law, effectively nullifying the effect of *sui generis* classification.

2 2 4 The literary-analogy problem

This doctrinal approach considers computer programs eligible for copyright protection because they are expressions *in writing* and, therefore, capable of being reproduced, translated or otherwise adapted in a manner similar to other literary work.

---

[239] This point is made and discussed in detail in chapter 3 below.

[240] The Copyright Act section 1(1) definition of literary work.

[241] See for example the list of Canadian, British and American cases applied in *Haupt t/a Softcopy v Brewers Marketing Intelligence (Pty) Ltd and Others* 20064 SA 458 (SCA) and *King v South African Weather Service* 2007 BIP 323 (T).

This approach is unsound and has been roundly criticised by scholars.[242] First, for the purpose of determining the nature and scope of copyright protection, little, if any, significance should be attached to the fact that computer programs exist in writing.[243] All types of work exist in writing, in the broad sense, in order to meet the requirement that the work has been reduced to a material form. The fact that a computer program looks like a literary work contributes nothing to, and should not be allowed to influence, the determination of the scope of protection or the application of the restricted acts.

Second, it is inappropriate to compare computer programs to literary work because a single analogous type of literary work cannot be identified. None of the traditional types

---

[242] See for example Cohen J E "Reverse Engineering and The Rise of Electronic Vigilantism: Intellectual Property Implications of "Lock-Out" Programs" 1995 *Southern California Law Review* 68 1091 at 1107-1111 where the author argues that "the classification of computer programs as literary works is inappropriate and breeds confusion" because it does not recognise the primarily utilitarian nature of the work. See also Menell P S "Tailoring Legal Protection for Computer Software" 1987 *Stanford Law Review* 39 1329 at 1365-7 where the author submits that copyright protection of computer programs as literary work is inappropriate and stifles innovation. Menell argues that "dominant firms (or anyone else) should not be able to 'lock up' an industry standard simply by expressing it in a unique way" and should, instead, rely on patent protection to focus on the unique utilitarian aspects of a program rather than its expression. See also Samuelson P et al. "A Manifesto Concerning the Legal Protection of Computer Programs" 1994 *Columbia Law Review* 94 2308 at 2357 where the authors submit that some attempts to accommodate protection of new technologies or sui generis forms of expression within the confines of traditional IP rights, lead to a cycle of under and over protection, resulting in improper extension of protection because, "although such boundary extension may solve one developer's immediate underprotection problem, it often hurts the market as a whole by overprotecting incremental technical innovations embodied in publicly distributed products." See contra Miller A R "Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?" 1993 *Harvard Law Review* 106 (5) 977 at 1201 where the author states that "by deciding that computer software is copyrightable and is to be treated as a literary work, a category in which no standardization exemption is recognized, Congress ruled out weighing contrary public interest considerations altogether." Miller's contention is comprehensively criticised and, it is submitted, correctly dismissed by Cohen in the work cited above. The primary mistake in Miller's reasoning is the underlying focus on the commercial impact that a derivative work might have, in future, on the interests of the copyright owner. This was not the appropriate question at hand. The impact of an infringing work is a matter that does not, or should not, determine the classification of the type of work or, particularly, the legality of intermediate copies which were made in order to read the work.

[243] Cohen *Southern California Law Review* 1107.

of literary works, such as novels, anthologies, dictionaries or plays, reflect all of the unique characteristic of a computer program. While some, such as a recipe book or directory, may be analogous to the instructional characteristic of a program,[244] others, such as a construction manual or engineering guide, better reflect the unique structural and sequential arrangement of instructions of a program. Therefore, if computer programs are treated as literary works, it must, at least, be treated as a *sui generis* type of literary work,[245] in which case the application of copyright principles to computer programs should depart from the manner in which these principles apply to other types of literary work. If this is not done, the risk of "importing into the analysis preconceptions of marginal relevance"[246] is realised. One striking example where this has happened, namely the translation of computer programs as a form of adaptation, is examined below in chapter 3. Two further examples, regarding incidental or intermediate reproduction, and computer programs as a compilation, are examined below in chapter 4.

Furthermore, it is unsafe to treat computer programs as analogous to literary work if there is no clear analogous type of literary work to which it must be compared, because, "each change in the statutory reference point conjures up a slightly different body of precedent and different variations."[247] While there may be an overlap in the application of some principles between, for example, a novel and a work of non-fiction, sharp differences exist in the application of copyright law principles between, for example, a directory of numbers and a collection of poems. To bundle computer programs with literary works in general, without a clear point of comparable reference, is to invite confusion and leave room for an *ad hoc* literary analogy.

---

[244] 1108.

[245] See Cohen *Southern California Law Review* 1108 where the author submits:

> "Traditional literary works exist on a continuum of protection; if computer programs are best characterized as literary works, it certainly would be reasonable to conclude that computer programs constitute a new endpoint on that continuum."

[246] Cohen *Southern California Law Review* 1109.

[247] Cohen *Southern California Law Review* 1110.

Third, the written appearance of the program should not be allowed to obscure the fact that, unlike all other examples of literary work, computer programs are not intended to communicate with humans. As Cohen puts it:

> "A computer program is, first and foremost, a series of instructions to the computer to execute a given task. The instructions themselves may be written or arranged with more or less creativity, **but that is not their primary significance**."[248]

In other words, unlike traditional literary works, computer programs are intended to communicate an instruction to a machine and are, therefore, not analogous to any traditional literary work. This means that computer programs represent a form of literary expression that is inherently different to all other forms of writing. For this reason, computer programs are published in a unique way, namely, object code. This form of writing is, unlike all other types of literary work, not humanly legible.[249] Where computer programs are treated as literary works, the extension of the exclusive right of adaptation, by translation or a change in code or notation, ignores the fact that these steps are necessary to extract the meaning of the work. Thus, while a traditional literary work may be read by a human conversant in the language in which it is expressed, this is impossible in the case of computer programs. As a result, the literary analogy prevents "programmers from consulting the copyrighted program [and] confer more protection on computer programs than on other copyrighted works."[250]

In addition to the general problems with a literary-works analogy discussed above, a further problem is apparent in the SA legislative context. Regardless of the reasons why the legislature chose to classify computer programs as a *sui generis* type of work,[251] the fact remains that it purposefully isolated this type of work from literary works. To continue to interpret copyright principles in SA according to precedent,

---

[248] 1108.

[249] In some cases, the object code may be read by a human with great effort and after extensive mathematical calculations, but, in the majority of cases and to the majority of computer programmers, object code is illegible. This point is explored further below in chapter 3.

[250] Cohen *Southern California Law Review* 1123-4.

[251] For a discussion of the legislator's intention and the legislative history in SA, see the discussion in paragraph 2 2 2 *et seq* above.

either local or foreign, that treat computer programs as analogous to literary work, is to ignore, and effectively render moot, the letter of the law.

Furthermore, in the absence of clear local precedent on the application of copyright principles, such as adaptation, translation or reproduction, *to computer programs* specifically, it is inappropriate to assume, as many local authors and some courts have done,[252] that these principles are applicable to computer programs to the same extent that they are applicable to literary works. In other words, the *sui generis* classification must have a more significant effect than just a classification – it mandates a unique interpretation of copyright principles and, at least, prohibits a literary-analogy approach.

Despite the above contentions, the counterargument may be made that the literary-analogy does not pose a problem for the correct interpretation of copyright law principles in the case of computer programs because the law has developed to give effect to the unique nature of this type of work. In other words, so the argument may go, the classification of computer programs as a type of literary work in some jurisdictions, or the literary analogy itself, is nothing more than a matter of convenience – it relies on the literary analogy only as a way of bringing computer programs into the sphere of copyright law without the need for extensive legislative amendment, but determines the scope of protection with the necessary sensitivity to its peculiarities.

However, this contention should fail, for the following reasons. First, while it may have been convenient to treat, or classify, computer programs as literary work in the early 1980s, today the software industry,[253] the nature of the computer programs and the creative process are all significantly different.[254] Therefore, to continue to treat computer programs as analogous to literary work maintains a perception that is no

[252] See the discussion in paragraph 3 2 *et seq.* below.

[253] See for example *Southern California Law Review* at 118-9 where the author points out that, "in recent years, research consortia, 'technology transfer' programs, and other joint ventures sponsored by corporate investors have become the preferred methods of innovation." Therefore, the need to access a work and the freedom to manipulate a program in order to discover its techniques and underlying ideas, have become industry standard practice.

[254] In these respects, see the discussion on cumulative innovation in paragraph 2 1 4 above.

longer accurate. In addition, the literary-analogy imposes restrictions on the exploitation of computer programs, such as the free decompilation of code for any reason, that is contrary to the idea/expression dichotomy and, consequently, violates the balance between the public and private interests in copyright protection.[255]

Second, and related to the first, is the fact that "modern-day copyright is substantially a creature of public policy"[256] and must, therefore, clearly serve the interests of all stakeholders in a direct way. It is no longer sufficient to argue that the indirect benefit to society, derived from an increase in the common knowledge, achieves a fair balance of interests in all cases. In the case of computer programs, it is shown that copyright law must, expressly, grant certain privileges to users in order to maintain a fair balance.[257] However, where the literary analogy continues to influence the formulation of an exception or limitation to the scope of protection in computer programs by, for example, relying on the interpretation of the rights of reproduction or adaptation in relation to literary works, it results in overly-narrow exceptions.[258]

Third, the argument that the problems with the literary-analogy, and the problems it creates, are not sufficient reason to suggest that computer programs should be treated differently from other types of literary works, does not take cognisance of the fact that computer programs are *unique*. Blending computer programs, to any extent, with literary work, "breeds confusion"[259] when it is treated as either literary work or perceived as quasi-literary work. In other words, even if it can be said that computer programs are *de facto* a sui generis type of work, because, for example, sufficient provision is made in law for its peculiarities,[260] the classification, or treatment of, computer programs as literary works has residual application.

[255] This point is argued in greater detail in chapter 5 below.

[256] Cohen *Southern California Law Review* 1115.

[257] See the discussion of a fair decompilation exception in chapter 5 below.

[258] In this respect see the discussion of the development of the fair use and fair dealing exceptions for decompilation in chapter 4 below.

[259] Cohen *Southern California Law Review* 1109.

[260] Such as, for example, the limits on the protection of non-literal or functional elements, discussed in chapter 4 below.

Where computer programs are classified as literary works, such as in the US or UK, it is not appropriate to argue that this has no impact on the development of copyright law in relation to computer programs. As shown below, the courts must, and do, give effect to the literary classification.[261] Where computer programs are not classified as literary works, such as in SA, but foreign law is used to interpret the application of copyright principles, the literary analogy has the same effect, and leads to an interpretation of copyright law that does not recognise the unique nature of computer programs.[262]

In light of these observations, it is submitted that an approach to computer programs influenced by a literary-analogy is inherently problematic from a legal positivistic point of view and from a technical point of view. It will be shown that, at least as far as the decompilation of computer programs is concerned, the problems associated with a literary-analogy represents a sufficient basis for advocating that copyright law should be developed in a different way.

However, this work does not rely on the literary-analogy alone as the basis for criticising the current legal approach to decompilation. In fact, the literary-analogy is secondary to the primary reasons why this work finds that decompilation should be regulated in a different, less restricted, manner. The primary bases, as will become clear, are the technical nature of the work and the fact that it is not published in an accessible form. In addition, this work finds sufficient impetus for an alternative approach to decompilation, in international copyright law and the three-step test, to support its findings, and the proposal for a pro-developmental exception to copyright law, based on public policy. In the next chapter, the first part of the primary basis for the findings in this work is developed, namely, the technical nature of decompilation, and the impact of the literary-analogy in SA copyright law regarding computer programs is illustrated. The purpose of this chapter is to determine if there is, or should be, sufficient room in SA copyright law, as a result of the *sui generis* classification of computer programs, to justify the act of decompilation.

---

[261] See the discussion of the US and UK cases in chapter 4 below.

[262] This point is discussed in detail in chapter 3 below.

Chapter 3

A Technical Analysis of Decompilation and Interpretation of the Applicable Restricted Acts

3 1 Decompilation

Traditional copyright "rules are grounded in real-world physics"[263] and rely, to a large extent, on the central tenet that copyright law bestows, and must safeguard, the right to make *physical* reproductions of a work.[264] This fact is evident in the material expression, or physical fixation, requirement for subsistence and, more importantly, the core principle of copyright law manifested in the idea/expression separation. It is, also, the primary dividing line between copyright law and patent law. Copyright deals with expressions, not ideas, and developed to prevent the making, and distribution, of unauthorised reproductions of that expression.[265]

Conversely, a computer program is a *sui generis* form of expression that does not comfortably fit within the traditional, physical, paradigm of copyright law. It exists in an intangible, digital medium,[266] is published primarily in encrypted, illegible form, and it is used in a manner that must, necessarily, involve the incidental reproduction of the work.[267]

This creates the potential for tension where jurisprudence, developed in relation to the restricted acts regarding analogue types of work, is applied to computer programs.

[263] See Weatherall K "IP in a Changing Information Environment" in: Bowrey K, et al. (eds.) *Emerging Challenges in Intellectual Property* (2011) 3. The author made this statement with reference to IP rights in general, but proceeded to deal with, and illustrate, this sentiment in relation to copyright protection.

[264] Weatherall "IP in a Changing Information Environment" 3.

[265] The development of copyright protection in computer programs in SA is discussed in this chapter further below. The development of copyright protection in computer programs in the USA and UK is analysed in chapter 4.

[266] With the exception of, for example, the preparatory work or program code drafted by hand.

[267] Every time a computer program is executed by a user, at least some, potentially substantial, part or parts of the program are reproduced by the computer into temporary storage and/or the processing unit(s) of the machine.

One example of this tension is the manner in which the restricted acts of reproduction and adaptation might, or do, apply to the act of decompilation. To illustrate this tension, and show why the impact of the literary analogy is inappropriate in the case of computer programs, the application of these rights is analysed in this chapter. This chapter submits that, if the act of decompilation is understood from a technical perspective, the restricted acts of reproduction and adaptation are not, or, in one case,[268] should not be, applicable and that decompilation should not be an infringing act.

In order to do so, the plethora of legal interpretive problems, which arise as a result the literary-analogy, are discussed in this chapter with a focus on the technical nature of decompilation, and in relation to SA copyright law. The legal position, and the consequences of the literary-analogy, in the USA and UK, are analysed in chapter 4.

3 1 1 Decompilation and reverse engineering

The process of decompilation is often described as a form of reverse engineering because it seeks to return the work to a state that resembles its original, source code, form, from which its constituent parts and construction may be apparent.

From a technical perspective, it is described as follows:

> "Reverse engineering is a critical set of techniques and tools for understanding what software is really all about. Formally, it is the process of **analyzing a subject system** to identify the system's components and their interrelationships and to **create representations of the system in another form** or at a higher level of abstraction. This allows us to **visualize** the software's structure, its ways of operation, and the features that drive its behavior. The techniques of analysis, and the application of automated tools for software examination, give us **a reasonable way to comprehend** the complexity of the software and to uncover its truth."[269]

---

[268] It is submitted further below that, if copying is interpreted widely to include use of the work to make a derivative work, the right to adaptation may be infringed. It is argued that this is the only circumstance under which copyright law principles do, in fact, prohibit decompilation and that this interpretation should be narrowed in order to justify the act of decompilation.

[269] Eilam *Reversing Secrets of Reverse Engineering* viii (original quotation omitted, emphasis added).

However, from a legal perspective, this process is foisted on the restricted act of adaptation because it looks like a process of reversing the initial translation process:

> "Decompilation involves copying a computer program and subsequently translating the program from a language that only a machine can understand to a language that a human can read with ease."[270]

The underlying mistake in this reasoning, namely, that compilation is a predictable process akin to textual translation, is the reason why some have observed that:

> "Fundamentally, the act of compilation does not change the instructions to the computer at all. It merely serves to translate the coded instructions from one computer language to another. The object code can also be **decompiled or disassembled to reveal the original source code**."[271]

As this chapter will show, this is an "attempt to stretch an inadequate statutory framework too far"[272] because, while the compilation process, which is under the control of the original programmer, may be a translation, the reverse process of decompilation is not, and does not deliver the source code.

Technically, decompilation is "an attempt to bring back the cow from the hamburger or the eggs from the omelet."[273] Despite the assumption of some that decompilation is an adaptation because it creates *versions* of code, this is shown to be technically

---

[270] Soobert *John Marshall Law Review* 105

[271] De Villiers *SALJ* 317. This view is directly based on the leap in logic by notable scholars who argue that because compilation may be viewed as a translation, decompilation must necessarily be viewed in the same light. See for example Bainbridge D I *Intellectual Property* 8ed (2010) 268 where the author submits that: "If the object code version of a program, produced by compiling or assembling a source code program, is later disassembled, to derive an assembly language version, that too falls within the meaning of making an adaptation." It is important to note the loose and technically inaccurate use of the term "version," by Bainbridge, to imply that decompilation or disassembly is a translation. See MacQueen H, et al. *Contemporary Intellectual Property Law and Policy* 2ed (2008) 140 at 4.34 and 159 at 4.67 generally for a discussion of adaptation and computer programs in the context of infringement. See also Lai S *Copyright Protection of Computer Software in the United Kingdom* 1ed (2000) 95 para 6.3 fn4 and para 6.4, where the author repeats the same assertion, without question, expressed by Bainbridge above.

[272] Soobert *John Marshall Law Review* 115.

[273] Eilam *Reversing Secrets of Reverse Engineering* 458.

untrue and not "highly analogous to translating a work of literature"[274] in any way. In other words, it is submitted that the text-oriented view of computer programs, which underpin the above reasoning, is problematic, from both a legal and technical point of view.

To understand the mismatch between the perception of decompilation in copyright jurisprudence and the technical nature of this process, it is necessary to clarify a number of key technical terms.

3 1 1 1 Pertinent technical terminology

As will become clear in this chapter, many of the terms created by copyright jurisprudence, such as, object code and source code or compilation and decompilation, are "only a convenient fiction"[275] used to explain in laymen's terms how computers work. In computer science these terms are not always discrete forms or distinct processes, which creates a number of problems[276] when copyright law attempts to maintain a clear divide between forms of expression or potentially infringing actions.

The reason why these terms were adopted in copyright law, and why they are useful in interpreting copyright law principles, has to do with the nature of software development. Computer programs are intended to direct the operation of a machine and must, therefore, communicate instructions in a very precise, mathematically reliable and predictable manner. This means that the instructions must be conveyed at a low level of abstraction, where the term 'low' describes a short distance between the human programmer's instruction and the machine, in a metaphysical sense.

---

[274] Bainbridge D I *Intellectual Property* 8ed (2010) 268.

[275] Touretzsky *Source vs. Object Code: A False Dichotomy - Essay submitted to the Court in the matter of Universal City Studios v. 2600 Magazine on July 25, 2000* 1, unpublished work, available at https://www.cs.cmu.edu/~dst/DeCSS/object-code.txt (accessed November 2019), recorded as trail exhibit BBE in *Universal City Studios v Reimerdes* (2000) 111 F. Supp. 2d 294 and *Universal City Studios v Corley* (2001) 273 F.3d 429.

[276] See Touretzsky *Source vs. Object Code* 1-3 for a list of some of the most common problems created by a separation of the object and source code as distinct legal terms. These points are discussed in more detail below.

In this context, the low level of abstraction describes the degree to which the instruction, written by the programmer, resembles the process the computer will carry out. In other words, low-level code, written in a low-level language, is less abstract. For this reason, low-level software is used as a collective, generic term to describe the infrastructure that directly, or very closely, operates the computer and includes low-level programming languages, assembly language, operating systems, and the development tools, such as the compiler program, used to create low-level code.[277] In other words, low-level software, also referred to as system software,[278] is "the layer that isolates software developers and application programs from the physical hardware."[279]

In legal parlance, low-level software is contrasted with high-level software, where the latter term describes programs written in a code that is more abstract. In other words, the instructions are written in a high-level language using programming terminology based on English language characters and requires little or often no mathematical calculation in order to be read and understood by a human. Thus, high-level software is at a greater metaphysical distance from the physical hardware and represents a greater degree of abstraction.

However, in order to direct the operation of a computer, all instructions must, eventually, be conveyed at the lowest level of abstraction namely, machine code, expressed in machine language. For the purpose of this overview, machine language represents instructions that are executed directly by the computer and are, therefore, beyond the lowest-point of low-level software. Machine code is read and carried out at the level of the computer architecture, by the physical hardware.

In the early days of computer programming, software engineers only worked at the low level of abstraction because the high-level infrastructure, such as high-level

[277] See Eilam *Reversing Secrets of Reverse Engineering* 9-10.

[278] This is a collective term for programs created at a low-level of abstraction, the programs used to create low-level code and the programs used to execute low-level code.

[279] Eilam *Reversing Secrets of Reverse Engineering* 10.

programming languages, universal compiler or interpreter programs and standardised or compatible operating systems, did not exist.[280]

Creating computer programs at this level of abstraction is laborious and difficult to perfect because the instructions are written in a form that more closely resemble machine code. In most cases this was done by writing directly in assembly language, the lowest level of abstraction. A number of examples of programs created in this manner are provided and discussed below in chapter 4.[281]

As computer science, and programming techniques, developed, and the tools for creating programs at a more abstract level became commonplace, it became the norm to create programs in high-level language and, thereafter, rely on other computer programs to convert the instructions into low-level code. Consequently, the terms source code and object code became known in copyright law as a means to distinguish between the highest level of abstraction, the source code, and the lowest level of abstraction namely, object code.

Similarly, the availability of computer programs which aid the process of decreasing the level of abstraction and transforming source code into object code, gave rise to the recognition of compilation and decompilation, in copyright jurisprudence, as a process akin to translating the humanly-legible instructions into machine-readable code. The origin of the terms object code, source code, compilation and decompilation, as they are understood for purposes of copyright law, is analysed in more detail in chapter 4 below. The technical nature of these terms is discussed, and illustrated, further below in this chapter.

From a technical perspective, it is clear that low-level and high-level software are not distinct opposites and, consequently, the subsequent terminology, developed in copyright law, such as object and source code or compilation and decompilation, are not distinct classes of code or directly-opposed processes.

[280] See Eilam *Reversing Secrets of Reverse Engineering* 10.

[281] See, in particular, the discussions in paragraphs 4 2 3 1 to 4 2 3 3.

The correct view is that low-level and high-level software are the end-points on a continuum of abstraction. This means that source and object code are relative terms which, depending on the particular case, mark certain points on the continuum of abstraction. However, these points are not fixed in every case. As discussed below, the object code of one program may be the source code of another program.[282] This is, in fact, common in modern computing where software is usually written at a high-level of abstraction and passed through a series of stages of abstraction, or so-called compilation, aided by a computer, in order to create low-level code.

Furthermore, the compilation of a program does not pass a set of instruction from the highest point of abstraction to the lowest. At best, compilation can be said to decrease, incrementally, the level of abstraction. The degree to which the compiled program may be seen as low-level software will depend on the manner in which it was compiled and the sophistication of the compilation process. In the same way that object code marks a point on the continuum of abstraction, compilation is a means to reach that point. But, as stated above, the object code of a particular program may be the source code for another program such as a further compiler or assembler program.[283]

This means that the source code and object code are not fixed opposites of the same original set of instructions.[284] As shown below, the process of decreasing the level of abstraction, which one may, for the sake of convenience, describe as compilation,

---

[282] See Touretzsky *Source vs. Object Code* 1 where the author uses the term "target code" to explain that either source or object code may be the 'target' of the program, i.e. the result that it produces and, therefore, the object code of one program may often be the source code of another.

[283] Touretzsky provides the following example:

> "For example, both early C++ compilers and the Kyoto Common Lisp compiler produced C code as their output. C++ (or Common Lisp) was the source language; C was the target language. This output was then run through a C compiler to produce symbolic assembler code as output. This then had to be run through yet another proram, the "assembler", to produce binary machine code that could be directly executed by a processor."

[284] See Touretzsky *Source vs. Object Code* 1 where the author submits that "'source' and 'object' are relative terms, not absolute categories." See also *Universal City Studios v Corley* (2001) 273 F.3d 429 at 439 where the court states that "object code can be placed on one end of a spectrum, and different kinds of source code can be arrayed across the spectrum according to the ease with which they are read and understood by humans."

does not turn source code into object code. At best, it can be said that compilation creates a low-level abstraction of the message which the programmer intended to convey.

Unfortunately, as shown below, copyright law does not recognise the relativity of object code and source code and, instead, applies these terms as specific forms of expression that are directly related. Consequently, the process of compilation is treated as a stage of development which *turns* source code into object code, rather than the more accurate view of compilation as an iterative process that *creates* object code. As a result, the law considers an attempt to reverse the process, by decompilation, as a form of reverse engineering that *returns* object code to its original source code.

This is technically inaccurate. To illustrate how the mismatch between the technical nature of computer programming and the perception of code and compilation in terms of copyright law, impacts on the legality of decompilation, an example may be useful.

3 1 1 2 Illustration of compilation and decompilation

As mentioned above, the technical reality is that software development, in the context of compilation, is a complex process. This makes it difficult to provide an accessible, precise and brief illustration of the difference between source code and object code or provide an accurate and reliable example of the difference between original source code and decompiled source code. The difficulty lies in the fact that an illustration of compilation does not provide an illustration of decompilation and, therefore, in order to understand both processes one must have reference to, inter alia, extensive program code, detailed explanations of the technical specifications of the tools used during the compilation and decompilation processes and a command of the programming languages used in the particular example.

To provide all of these details, within the confines of this work, is not possible. Therefore, in order to illustrate the salient technical points, a suitably reliable and accessible example has been selected from written expert testimony[285] submitted to the district court of New York in the case of *Universal City Studios v Reimerdes*[286] and cited with approval by the court of appeals in *Universal City Studios v Corley.*[287] More extensive and detailed examples of sample programs may be found in the accompanying online resources to the seminal work on decompilation techniques.[288]

The original program below, called FACT, is a simple calculator for determining the value of 5!, i.e. the 5 factorial, or the result of multiplying all the values below five in successive order.

The program code is reproduced here in table format, retaining the original layout and spacing as far as possible. In order to facilitate ease of cross-referencing between the programs, additional line numbers have been added in the left column. One key instruction is highlighted below, in each of the figures, which will be referred to later.

Figure 1: FACT in original source code written in the C programming language

1	#include <stdio.h>
2	void main(int argc, char *argv[]) {
3	int i, result;
4	result = 1;
5	for (i = 1; i<6; i++) {
6	result = result * i;
7	}
8	printf (“Result is: %d.\n”,result);
9	}

[285] The example below was provided by David Touretzky, Research Professor in the Computer Science Department and the Center for the Neural Basis of Cognition at Carnegie Mellon University.

[286] *Universal City Studios v Reimerdes* (2000) 111 F. Supp. 2d 294.

[287] *Universal City Studios v Corley* (2001) 273 F.3d 429. The program code, reproduced in this work, was published as part of a draft article. See Touretzsky D S *Source vs. Object Code*, recorded as trail exhibit BBE in both cases.

[288] See Eilam *Reversing Secrets of Reverse Engineering* xxviii.

Figure 2: FACT in assembly language

```
        .file   "fact.c"
gcc2_compiled.:
        .global .umul
.section        ".rodata"
        .align 8
.LLC0:
        .asciz  "Result is:  %d.\n"
.section        ".text"
        .align 4
        .global main
        .type    main,#function
        .proc   020
main:
        !#PROLOGUE# 0
        save    %sp, -120, %sp
        !#PROLOGUE# 1
        st      %i0, [%fp+68]
        st      %i1, [%fp+72]
        mov     1, %o0
        st      %o0, [%fp-24]
        mov     1, %o0
        st      %o0, [%fp-20]
.LL3:
        ld      [%fp-20], %o0
        cmp     %o0, 5
        ble     .LL6
        Nop
        b .LL4
         Nop
.LL6:
```

31	ld [%fp-24], %o0
32	ld [%fp-20], %o1
33	call .umul, 0
34	Nop
35	st %o0, [%fp-24]
36	.LL5:
37	ld [%fp-20], %o0
38	add %o0, 1, %o1
39	st %o1, [%fp-20]
40	b .LL3
41	Nop
42	.LL4:
43	sethi %hi(.LLC0), %o1
44	or %o1, %lo(.LLC0), %o0
45	ld [%fp-24], %o1
46	call printf, 0
47	Nop
48	.LL2:
49	Ret
50	Restore
51	.LLfe1:
52	.size main,.LLfe1-main
53	.ident "GCC: (GNU) 2.95.2 19991024 (release)"

Figure 3: FACT in binary code expressed in hexadecimal values

1	0000000 7f45 4c46 0102 0100 0000 0000 0000 0000
2	0000020 0001 0002 0000 0001 0000 0000 0000 0000
3	0000040 0000 0234 0000 0000 0034 0000 0000 0028
4	0000060 0008 0001 002e 7368 7374 7274 6162 002e
5	0000100 7465 7874 002e 726f 6461 7461 002e 7379

6	0000120 6d74 6162 002e 7374 7274 6162 002e 7265
7	0000140 6c61 2e74 6578 7400 2e63 6f6d 6d65 6e74
8	0000160 0000 0000 9de3 bf88 f027 a044 f227 a048
9	0000200 9010 2001 d027 bfe8 9010 2001 d027 bfec
10	0000220 d007 bfec 80a2 2005 0480 0004 0100 0000
11	0000240 1080 000c 0100 0000 d007 bfe8 d207 bfec
12	0000260 4000 0000 0100 0000 d027 bfe8 d007 bfec
13	0000300 9202 2001 d227 bfec 10bf fff2 0100 0000
14	0000320 1300 0000 9012 6000 d207 bfe8 4000 0000
15	0000340 0100 0000 81c7 e008 81e8 0000 0000 0000
16	0000360 5265 7375 6c74 2069 733a 2020 2564 2e0a
17	0000400 0000 0000 0000 0001 0000 0000 0000 0000
18	0000420 0400 fff1 0000 0001 0000 0000 0000 0000
19	0000440 0400 fff1 0000 0000 0000 0000 0000 0000
20	0000460 0300 0003 0000 0008 0000 0000 0000 0000
21	0000500 0000 0002 0000 0000 0000 0000 0000 0000
22	0000520 0300 0002 0000 0017 0000 0000 0000 0000
23	0000540 1000 0000 0000 001d 0000 0000 0000 0000
24	0000560 1000 0000 0000 0024 0000 0000 0000 0078
25	0000600 1200 0002 0066 6163 742e 6300 6763 6332
26	0000620 5f63 6f6d 7069 6c65 642e 002e 756d 756c
27	0000640 0070 7269 6e74 6600 6d61 696e 0000 0000
28	0000660 0000 003c 0000 0507 0000 0000 0000 005c
29	0000700 0000 0209 0000 0000 0000 0060 0000 020c
30	0000720 0000 0000 0000 0068 0000 0607 0000 0000
31	0000740 0061 733a 2057 6f72 6b53 686f 7020 436f
32	0000760 6d70 696c 6572 7320 342e 3220 6465 7620
33	0001000 3133 204d 6179 2031 3939 360a 0047 4343
34	0001020 3a20 2847 4e55 2920 322e 3935 2e32 2031
35	0001040 3939 3931 3032 3420 2872 656c 6561 7365
36	0001060 2900 0000 0000 0000 0000 0000 0000 0000
37	0001100 0000 0000 0000 0000 0000 0000 0000 0000

38	0001120 0000 0000 0000 0000 0000 0000 0000 0001
39	0001140 0000 0003 0000 0000 0000 0000 0000 0034
40	0001160 0000 003d 0000 0000 0000 0000 0000 0001
41	0001200 0000 0000 0000 000b 0000 0001 0000 0006
42	0001220 0000 0000 0000 0074 0000 0078 0000 0000
43	0001240 0000 0000 0000 0004 0000 0000 0000 0011
44	0001260 0000 0001 0000 0002 0000 0000 0000 00f0
45	0001300 0000 0011 0000 0000 0000 0000 0000 0008
46	0001320 0000 0000 0000 0019 0000 0002 0000 0002
47	0001340 0000 0000 0000 0104 0000 0080 0000 0005
48	0001360 0000 0005 0000 0004 0000 0010 0000 0021
49	0001400 0000 0003 0000 0002 0000 0000 0000 0184
50	0001420 0000 0029 0000 0000 0000 0000 0000 0001
51	0001440 0000 0000 0000 0029 0000 0004 0000 0002
52	0001460 0000 0000 0000 01b0 0000 0030 0000 0004
53	0001500 0000 0002 0000 0004 0000 000c 0000 0034
54	0001520 0000 0001 0000 0000 0000 0000 0000 01e0
55	0001540 0000 0052 0000 0000 0000 0000 0000 0001
56	0001560 0000 0000
57	0001564

Figure 4: The result of decompilation, by disassembly of the binary file in figure 3 above, expressed in low-level (assembly) language.

1	section .text			
2	main()			
3	0:	9d e3 bf 88	save	%sp, -120, %sp
4	4:	f0 27 a0 44	st	%i0, [%fp + 68]
5	8:	f2 27 a0 48	st	%i1, [%fp + 72]
6	c:	90 10 20 01	mov	1, %o0
7	10:	d0 27 bf e8	st	%o0, [%fp - 24]

8	14:	90 10 20 01	mov	1, %o0
9	18:	d0 27 bf ec	st	%o0, [%fp - 20]
10	1c:	d0 07 bf ec	ld	[%fp - 20], %o0
11	20:	80 a2 20 05	cmp	%o0, 5
12	24:	04 80 00 04	ble	0x34
13	28:	01 00 00 00	nop	
14	2c:	10 80 00 0c	ba	0x5c
15	30:	01 00 00 00	nop	
16	34:	d0 07 bf e8	ld	[%fp - 24], %o0
17	38:	d2 07 bf ec	ld	[%fp - 20], %o1
18	3c:	40 00 00 00	call	0x3c
19	40:	01 00 00 00	nop	
20	44:	d0 27 bf e8	st	%o0, [%fp - 24]
21	48:	d0 07 bf ec	ld	[%fp - 20], %o0
22	4c:	92 02 20 01	add	%o0, 1, %o1
23	50:	d2 27 bf ec	st	%o1, [%fp - 20]
24	54:	10 bf ff f2	ba	0x1c
25	58:	01 00 00 00	nop	
26	5c:	13 00 00 00	sethi	%hi(gcc2_compiled.), %o1
27	60:	90 12 60 00	or	%o1, gcc2_compiled., %o0 !
28	gcc2_compiled.			
29	64:	d2 07 bf e8	ld	[%fp - 24], %o1
30	68:	40 00 00 00	call	0x68
31	6c:	01 00 00 00	nop	
32	70:	81 c7 e0 08	ret	
33	74:	81 e8 00 00	restore	

3 1 1 3 Comparison and analysis of the program illustration

With the aid of the above examples, a number of brief observations may be made which will be discussed in more detail later in this chapter.

First, the text in figure 1, the original source code, represents the high-level instructions. Figure 2 is a lower-level abstraction thereof, expressed in assembly language, created by a process of compilation. Thus, according to copyright law, figure 1 is source code and figure 2 is object code. However, clearly the contents of figure 2 is not in binary code and, therefore, not *directly* executable by the machine.[289]

Figure 3 expresses the instructions in binary code, in the hexadecimal format. Thus, for the purpose of figure 3, the contents of figure 2 is source code, not object code. This means that figure 2 is, if the strict separation of classes of code in copyright law is adhered to, simultaneously object code and source code.

This should make it clear that the process of compilation does not translate source code into object code. From a technical perspective, figure 1 is the source code and figure 3, not figure 2, is the object code. Therefore, there is no translation from figure 1 into figure 3. At best, figure 3 is a translation of figure 2 which is, in turn, a translation of figure 1. Thus, it is technically incorrect to view compilation as a process of translating source code into object code. As will be shown below, it is also improper to consider the process of compilation, at each stage described above, as the making of a *translation* in terms of copyright law.

Second, even on close comparison of the texts in figures 1 and 2, it is clear that during compilation *no copying of code*, in the sense of literal reproduction, occurs from the original source code into the object code. This means that, even if decompilation is considered a reversal of the compilation process, the decompilation result (figure 4) cannot, technically, be a copy of the literal instructions in the original source code because the work that was used to create the decompilation result, the object code in

[289] It may be executed without the need for further compilation if, for example, another intermediate program, called an interpreter, is used to generate binary code in real-time while the assembly code is executed. However, it does not mean that assembly code is technically object code.

figure 3, was not a copy of the original source code. This means that, where copyright law is applied to decompilation, the term reproduction cannot refer to similarities between the decompiled code and the original source code because no copying took place. Consequently, if decompilation is to be considered an act of making a reproduction of the *original source code*, the term reproduction must be applied widely, to encompass reproduction, of a reproduction, by indirect copying or derivative use.[290]

Third, the examples above illustrate the decreasing levels of abstraction. A comparison of the key instruction in figure 1 and 2, both of which are humanly legible, make this clear. In line 5 of figure 1, the programmer wrote the instruction “i<6”. This expresses his intention that the computer program will calculate the factorial of all numerical values that are **smaller than 6**. The instruction is written in a FOR loop, a programming technique used to create a repeatable instruction with inherent variability in a short amount of code, which requires less processing power.[291]

In other words, in lines 5 to 6 of figure 1, the instruction is given that, starting with the number 1 (the starting value of the variable “result”), the computer should multiply it with the value attributed to the variable *i*. On the first cycle, the value of *i* is equal to 1 and the result is, therefore, 1. On the second cycle, the value of *i* has been increased by the number of times the FOR loop cycle has been carried out, in this case once. Thus, the value of *i* is now equal to 2 and the most recent value of “result” is still 1. The calculation is, therefore, 1x2 and the result is 2. On the third cycle, the value of *i* is three and the most recent value of “result” is 2, the calculation is 2x3 and the result is 6. This process is continued, until, as the programmer intended, the value of *i* is smaller than 6, i.e. equal to five. At this point, the FOR loop stops, and the next line of code is executed. This line, the “printf” function, displays the result of the last calculation, namely, the value 120, which is the 5 factorial, or the value of 5!.

The decision to write the instruction “i<6” is a creative choice of the programmer, and influenced the way in which the FOR loop was written. The programmer could have

[290] This argument is discussed in more detail below.

[291] It is called a FOR loop because it is a set of instructions that should be repeated (i.e. a loop) until a set condition is met (i.e. for as long as a condition is not met).

achieved the same result by, for example, using the instruction "i=5" instead or, alternatively, written the FOR loop to start with the value of *i* equal to 5 and subtracting, instead of adding, the number of times the FOR loop has been carried out, until *i*=0. In both cases, the program would deliver the same result. The programmer's decision, in this case, is primarily a choice influenced by his preference for, or habit, of writing a FOR loop that works upward in value rather than down or, perhaps, a decision to make the instruction as obvious, reliable or short as possible.

However, his creative expression, in this case, is not conveyed by the object code because the rules of the programming language dictate that a FOR loop will not be carried out until a value is smaller or larger than a variable but, instead, terminate when the variable is *equal* to a variable. For this reason, the "i<6" instruction is expressed, in assembly code form, as "cmp %o0, 5" (line 25 of figure 2), which compares the result to the value 5. Thus, the object code of this instruction, 80a22005 (line 10 in figure 3), expresses only the fact that process is carried out 5 times. It does not, in any way, suggest that the original programmer intended the process to be carried out **less than 6 times**.

In mathematical terms this difference is immaterial, but in legal terms it is of significant importance. It illustrates the fact that the expression of the idea is not present in the object code. The way in which the programmer wrote the FOR loop, to operate upward in value, is completely removed from the object code.

Thus, when the object code is decompiled, the source code is *not revealed.* All that can be learned from the decompiled code is that a factorial of 5 is calculated. This is evident in line 11 of figure 4, which contains the same "cmp %o0, 5" instruction.[292] In other words, the decompilation result tells the reader that the original programmer had the idea to calculate 5 factorial, but it does not tell him that the original source code

---

[292] The fact that these lines are exactly the same, illustrate that the decompilation process achieved a very close approximation of the instruction in the original assembly language. Considering the very basic nature of this instruction, it is not surprising. However, if the instruction was more complex, the result would be less similar and the person who reads the decompiled code would have to interpret several lines of code in order to understand the idea which could have given rise to the program.

used the “i<6” instruction or any of the instructions that were written to carry out the calculation in this manner.

Consequently, it is clear that decompilation does not reproduce, and therefore cannot reveal source code, or the creative expression. The only thing it *reveals*, is the idea behind that instruction, in broad terms.

This should also make it clear why decompilation is a justifiable form of use. As the author of the above code points out, “the fact that the constant is 6 in the C code [figure 1] and 5 in the assembly language code [figure 2] is the sort of thing one can learn only by looking at the assembly language code.”[293]

Therefore, in order to learn that the program calculates the 5 factorial, one must be able to read the assembly language version. Where this is not published, as is common practice, the only way to learn how the program works, is to decompile it. As shown above, this process reveals how the program works in broad terms, but it does not provide access to the source code instructions or the assembly code.

One must also keep in mind that the example provided above is a very simple program and the correlation between the source code, object code and decompiled code is, consequently, very close. In practice, the significantly more complex instructions of even basic computer programs will mean that the transformation which occurs during compilation will be more substantial and, consequently, the correlation between the object code and the decompiled code, significantly less reliable or easy to understand. Nevertheless, it should be clear that, if the decompilation result of a simple program such as FACT does not reveal the creative expression of the original programmer, the result of decompiling a more complex program would certainly deliver a result that is even less relatable to the original source code.

This should make the point clear that decompilation is a process to discover the ideas of a program and not its expression. Therefore, it is not a broad sentiment or a trick of terminology to state that decompilation *does not reproduce code* – it is a technical fact.

[293] Touretzky *Source vs. Object Code* 3.

This has implications on the interpretation of copyright law principles, particularly in relation to the restricted acts of reproduction and adaptation, which are discussed further below.

The fourth observation is less evident in the above illustration of program code and relates to the technical nature of the process by which decompilation is carried out. It is clear that the only work created by a human is the source code, figure 1 above. The assembly code and the binary code (figures 2 and 3), are created by a computer. This compilation process, as explained above, involves the use of one or more computer programs, described as compilers, assemblers or interpreters, depending on the situation. In order to start this process, the source code is loaded into the memory of the computer, from where it is accessed, read and interpreted in order to create further code, which may take the form of assembly code, binary code or a variety of intermediate forms of lower, or less, abstraction.

Therefore, the whole of the original source code is copied, literally, into the temporary storage of the computer. In some cases, where the process is not compilation but interpretation, only portions of the code may be copied at a time. This difference is immaterial for present purposes. The point is that the code is literally copied into temporary storage. This is necessary because, as discussed above, the process of compilation must read and interpret the source code in order to draft a set of instructions at a lower level of abstraction.

In this context, the process of decompilation must be viewed. The primary purpose of decompilation is to increase the level of abstraction to the point where it can be read and understood by humans. Thus, like the process of compilation, decompilation is carried out by sophisticated computer programs designed for this purpose. This means that the object code of the original program, the only copy which a user will have available to him, must be copied into the temporary storage of the computer that will carry out the decompilation process. Once the process is complete, the decompiler program delivers a result that is, usually, expressed in assembly language and, therefore readable. At this point, the temporary storage is cleared and the copy of the object code removed. Similar to the process of interpretation, the decompilation process may be carried out in a piecemeal fashion, but the difference is not important.

In all cases, the computer requires that a literal reproduction of the code is made while the decompilation process is ongoing.

This means that, from a copyright law perspective, decompilation does involve the making of a reproduction of the object code. As will be shown below in the analysis of US and UK case law in chapter 4, this is the basis of infringement by decompilation. In case law, such reproductions are interchangeably described as temporary, incidental or, most commonly, intermediate copying.

Considering that general copyright law principles extend the meaning of reproduction to incidental copying and, in some case, the provisions of copyright law explicitly prohibit the making of intermediate copies of a computer program, the primary basis of a decompilation prohibition has been established. This is the reason why, as shown below in chapter 4, the majority of courts considered decompilation as a form of reproduction *per se*. From a technical perspective, this is correct. However, in light of the fact that decompilation is necessary in order to read the work and understand its ideas, this work, further below in this chapter, questions whether it is an appropriate interpretation of the reproduction right.

In light of the above observations about the technical nature of decompilation, the discussion now turns to an analysis of the legal position in SA copyright law regarding decompilation, and examines in greater detail the points made above regarding translation, reproduction and adaptation of code.

3 2 The restricted acts and decompilation in SA copyright law

Prior to the 1992 Amendment Act, computer programs qualified for protection as literary works regardless of its linguistic status.[294] In other words, South Africa adhered to the classic duality which accepted that computer programs may exist in one of two

[294] This view was expressed in *Apple Computer v Rosy t/a SA Commodity Brokers (Pty) Ltd* at 136, where the court accepted a wide meaning of the word translation and elected to view the source code and object code as expressions of the same work in different languages.

forms,[295] namely, source code and object code, and that both shall be eligible for protection.

From this, it followed that the object code of a program is an adaptation, in the form of a translation (albeit an automatically-generated translation, i.e. a computer-generated work in the strict sense) of the source code, which is made for the exclusive purpose of changing the humanly-legible source code into “machine readable”[296] object code.

The same position was re-applied, without any analysis, to computer programs as a *sui generis* type of work after the 1992 Amendment Act. Consequently, the construction of computer programs as literary works, and the concomitant view of object code as an adaptation or reproduction by translation of the source code, was adopted into post-1992 jurisprudence.[297]

However, the mistake is not entirely due to these judgments. The Copyright Act, post 1992, provides that an adaptation of a computer program includes “a version of the program in a programming language, code or notation different from that of the program.”[298] Therefore, as described above, the view developed among SA scholars that decompilation amounts to an infringing act, even though this has, to date, not been supported by local case law.

Consequently, so the logic goes, if the process of compilation is considered equal to the making of a translation, the definition of adaptation is clearly capable of encompassing, and thereby preventing, the unauthorised making of a reverse translation, i.e. decompiling the work. Thus, the restricted act of adaptation is extended to apply to decompilation in principle.

---

[295] See the discussions of the origins of computer programs as copyright work in the US and UK, below.

[296] *Haupt t/a Softcopy v Brewers Marketing Intelligence (Pty) Ltd and Others* 2004 BIP 207 (C) 219.

[297] See for example *King v South African Weather Service* at [9.5] where the court accepts as fact that the computer programs are “programs (ie source codes and executable files)” for purposes of copyright law or *Haupt t/a Soft Copy v Brewers Marketing Intelligence* 2004 BIP 207 (C) at 219 where the court relies on the observation that object code is a textual set of instructions and thus qualified for protection in the same instance as the source code which gave rise to it.

[298] Section 1(1) definition of adaptation.

By implication, it follows that the right to reproduction might also be applicable to decompilation if the process involved substantial copying. In addition, as shown above, the act of reproduction is also relevant regarding the intermediate copying of code during decompilation.

Thus, to summarise the prevailing views mentioned above, the restricted acts of adaptation or reproduction are said to be infringed by the process of decompilation in one, or more, of the following ways: (1) adaptation of the original source code or the object code by making a translation of the object code; (2) adaptation of the object code by altering the programming language, code or notation of the work; (3) reproduction of the original source code by decompiling the object code so that the original source code is revealed; (4) reproduction of the original source code or the object code by temporary or intermediate copying as part of the technical process.

In order to arrive at the purpose of this chapter, expressed above, namely, to illustrate the mismatch between copyright law and the technical reality of decompilation, each of the contentions listed above are discussed, in turn, below.

3 2 1 The restricted act of adaptation

The first two contentions are based on the restricted act of adaptation which may be infringed by the act of decompilation in two different ways, namely, translation and alteration of the language, code or notation, respectively. Each of these arguments is addressed individually.

3 2 1 1 Adaptation by translation

The traditional copyright principle of translation implies that the literary expression has been converted from one language to another. Because computer programs are also written in languages, albeit programming languages, and the source code and object code are viewed as different versions of the same literal expression, it follows that computer programs are translated when it is compiled.

During the early development of copyright protection for computer programs, this legal construction was essential in order to provide protection for computer programs under copyright law. Without it, only the source code would qualify for protection but not the object code. This argument was at the heart of some of the earliest cases on software copyright.[299]

From this point of view, the law developed to reflect the conclusion that, if compilation is translation, the reverse process thereof, namely, decompilation, must also involve the act of making a translation.

This perpetuated the notion that computer programs may be treated as literary works and, consequently, should be protected regardless of the language in which it is expressed. Therefore, the act of translation, by decompilation, amounts to an infringing act in the absence of an adaptation license or some other authorization.

It is submitted that this view is incorrect and at odds with the technical reality of decompilation, as illustrated above. It is argued below that the meaning of translation in copyright law, which was developed with reference to literary work, is incompatible with computer programs and should no longer be applicable, for the following reasons.

3 2 1 1 1 Message or meaning

From a technical perspective, the process of decompilation does not "reveal"[300] the source code in the same way that a translation of a journal article from German to English will reveal its *message* and its *meaning* to an English reader.[301] In the above illustration, for example, the instruction "i<6" was the message, while the meaning of this message is to repeat the process 5 times. It has been shown that the message is not conveyed by the object code because it is not contained therein and can, therefore, not be revealed by decompilation. As shown above, only the meaning, namely, the 5-times repetition, may be discovered by studying the decompilation result.

Therefore, the translation analogy can, at best, be applied to suggest that decompilation will provide clues about the *meaning* that the source code wished to

[299] See for example the discussion of the early cases in paragraphs 4 2 3 1 to 4 2 3 4 below.

[300] De Villiers *SALJ* 317.

[301] Bainbridge submits that decompilation only "unlocks the ideas and techniques contained in the object code". See Bainbridge *Intellectual Property* 268.

convey to the computer. It does not, however, provide the *message*. Decompilation only suggests what that message could have been, in an abstract manner. Thus, unlike translation of a literary work, decompilation does not provide the message and the meaning of the original work. Decompilation is manifestly inaccurate when compared to textual translation as a means to convey meaning and message. This is why "the decompilation process is expensive, laborious and resource-intensive,"[302] because it guesses at the message and the meaning of a work based solely on a derivate work, namely, the object code, in which the message and meaning has been altered to suit a machine. As the above illustration made clear, when source code is compiled only the meaning of the message is conveyed, in a less abstract manner, without any regard for maintaining the integrity of the original message. The purpose of compilation is not to preserve the message but to express the meaning thereof in a way that a computer will be able to understand and carry out. For this reason, in the above illustration, the message was abandoned at the first stage of compilation, into assembly language, and only the meaning was conveyed in the object code.

Furthermore, as shown above, decompilation is a computer-aided *approximation* exercise, carried out by a computer program. It is technical in nature and does not heed the rules of translation in the same manner as textual translations. Instead, it seeks to create a set of instructions, in a higher-level code, that could deliver the same object code, based on the known rules of a particular programming language, common programming techniques and standard, or commonly used, source code instructions. Therefore, in the same way that compilation does not translate source code into object code, decompilation does not translate object code into source code. Like compilation, decompilation creates a new set of instructions based on the message conveyed by the code. However, unlike compilation, decompilation is not precise and delivers only an approximation.

Therefore, although the outcome of the decompilation process will convey some meaning, to the extent that it suggests how a hypothetical programmer may have initiated the creation of the object code by drafting the source code, the result of the

[302] Jooste C and Karjiker S "Intellectual Property Law in the Digital Environment (EIP Law)" 414. See also Soobert *John Marshall Law Review* 115.

decompilation process does not deliver a translation of the original source code or the object code.[303] Instead, the process of decompilation delivers a "simulacrum" or approximation of the source code,[304] which is only "useful to obtain missing knowledge, ideas, and design philosophy".[305]

In other words:

> "In the software world reverse engineering boils down to taking an existing program for which source-code or proper documentation is not available and **attempting** to recover details regarding it's design and implementation."[306]

Thus, there are fundamental differences between textual translation and the nature of decompilation which, it is submitted, makes the term translation incompatible with decompilation. The meaning of "translation" in copyright law cannot be interpreted to mean conjecture. While textual translation may involve some degree of appraisal of the message or the meaning the author wished to convey, the whole of the translation process is not a speculation exercise. Furthermore, textual translation of a work has at its disposal the actual, original message and may read this in order to convey the meaning thereof in another language or format. Conversely, decompilation has available to it only the meaning of the work and must attempt to construct what the message could have been. This is, clearly, not akin to translation in the usual or legal sense of the word.

3 2 1 1 2 Functional or literal

The second reason why decompilation cannot be equated with translation of a literary work is because this view ignores the fact that the scope of copyright in computer programs is restricted to a set of instructions which directs a computer to "bring about a result."[307] Translation of a literary work is not carried out to change the purpose of

---

[303] Lee D "Reverse Engineering of Computer Programs under the DMCA: Recognizing a 'Fair Access' Defence" (2006) 10 *Marquette Intellectual Property Law Review* 538 541-2.

[304] Samuelson P and Scotchmer S "The Law and Economics of Reverse Engineering" (2001) 111 *Yale Law Journal* 1575 1608.

[305] Eilam *Reversing Secrets of Reverse Engineering* 3.

[306] xxiv (emphasis added).

[307] The Copyright Act section 1(1) definition of 'computer program'.

the work – it is merely a literal change. Compilation, on the other hand, must make something which is literal text into directly functional instructions for a computer. As discussed above, it is a process designed to decrease the level of abstraction.

The process of compilation is a necessary step in the development of a computer program to enable it to bring about a result *directly*, by the operation of the machine. This means the code must, in the final form, exist in a state that a computer can read. This requires that the code adhere to a fixed set of rules for mathematical calculations according to the specific manner these calculations are carried out by a specific machine. In other words, the code must meet the expectations of the computer.

While the set of instructions has not been compiled it is, at best, capable of directing the operation of a computer *indirectly*, on condition that it is either compiled into object code or executed with the aid of another computer program. Therefore, the process of compilation *creates* a set of instructions.

Technically, during the process of compilation, the high-level representation of the instructions to the machine, namely, the source code, is used to generate "an intermediate representation of the source program that attempts to classify exactly what the program does, in *compiler-readable form*."[308] This intermediate step is then used by the compiler program to draft a lower-level representation of the instructions. During this process, the original instructions are interpreted and new instructions are written according to, inter alia, the rules of the target language, to optimize the final object code. Thus, the process of compilation is not a translation – it is a reformulation of the initial instructions, with additions and alterations, into machine-readable format.

Conversely, when an attempt is made to reverse this process, the programmer has no idea about which language the source code was written in, how it was constructed, how, or by what means, it was compiled or which changes the programmer made to any part of this process. All that is at his disposal is technical knowledge about how compilation usually works. Based upon this information, the decompilation program presumes what the nature of the above factors could have been. This is "the key

[308] Eilam *Reversing Secrets of Reverse Engineering* 458.

difference between compilers and decompilers that often makes decompilation a far more indefinite process. Decompilers read machine language code as input, and such input can often be very difficult to analyze"[309] because it was written to direct the machine to produce a result, not to convey a message to a human.

Furthermore, as shown above, the nature of compilation makes it clear that source code and object code are "relative terms."[310] They merely describe the start and the end of a particular compilation process and represent the input and output of a single transformation cycle which may relay the intention of the programmer, expressed in a higher level language, into a lower-level language that is either used as the source code for a further transformation cycle or for the final assembly cycle.[311] The development of software may involve a repetition of this process, several times, in order to reach the level of sophistication and functional reliability that is desired. This process is best described as incremental development, but at no point is it equal to the making of a translation. Source and object code are best viewed as indicators of the level of abstraction of the idea on a continuum from human expression to machine instructions.

3 2 1 1 3 External influences during compilation

In the case of literal translation, the content of the translation is entirely dependent on the content of the original work. Variations in the message or meaning are only influenced by the rules of human language and the translator's level of skill or command of the relevant languages.

Conversely, during compilation the object code is not wholly determined by the source code. It is the result of a technical process, aimed exclusively at performing a function, which is influenced by a variety of external factors. These include, in the first place, the source code. However, the form, format and set of instruction embodied in object code are also influenced by, and in some instances determined by, other parameters such as the settings of the compiler program and the plethora of peculiarities unique

[309] Eilam *Reversing Secrets of Reverse Engineering* 458.

[310] Touretzsky *Source vs. Object Code* 4.

[311] 4. See also paragraph 3 1 1 1 above.

to the hardware and software environment in which the program must operate.[312] An example of this fact was provided above where it is shown that the compilation process elected to express the meaning of “i<6” in a different way, using the constant value 5 instead.

This is the reason why decompiled code is useful to discover structural efficiencies and techniques that may be useful to develop or improve other products,[313] because it provides information about *how* the program works and the technical restrictions it had to meet, rather than reveal the instructions written to enable those functions.

For example, during the compilation process, the “lexical, syntactic, and semantic”[314] structure and composition of the source code is analysed and interpreted. These elements represent, inter alia, the efficiencies, ergonomics and logical structure intended by the programmer. The compiler acts upon these instructions and determines how the object code should be written. In the above example, the compiler program changed the method of calculating the termination value of the FOR loop because it was a more direct, reliable way of determining the final value. It is more efficient, and more secure, to use a fixed value (5 in this case) as reference point, rather than the value of a calculated variable (less than 6).

Furthermore, the compilation process is itself dependent on certain restrictions and efficiencies which the programmer either seeks to exploit or avoid. Therefore, the object code is influenced by the “built-in biases”[315] of the computer programming

---

[312] See Velasco J “The Copyrightability of Nonliteral Elements of Computer Programs” (1994) 94 *Columbia Law Review* 242 245-7 in relation to the compilation process. See also, for example, the technical variations and tools applicable in this process listed by Gream M “A brief note on the reverse engineering of protected computer programs from a UK perspective” (2003) http://matthewgream.net/Professional/IntellectualProperty/note_decompile-comp-prog.pdf (accessed November 2019).

[313] Eilam *Reversing Secrets of Reverse Engineering* 8.

[314] Lin D S, Sag M and Laurie R S “Source Code versus Object Code: Patent Implications for the Open Source Community” (2011) 18 *Santa Clara High Technology Law Journal* 235 238-9.

[315] Davidson D M “Protecting Computer Software: A comprehensive analysis” (1983) 23 *Jurimetrics* 337.

language which will “determine the logic and design of the program.”[316] Consequently, the compiled work will contain instructions which are unique and symbolic representations, which are not a reworking, or ‘translation’ of the source code or, necessarily, representative of the intellectual endeavour of the programmer. In the above illustration, this is evident in figure 2 where the “i<6” instruction is expressed as “cmp %o0, 5”. The “cmp” part of this instruction is a compare command. It is a standard phrase or term which means that a value is to be *compared* with the constant value contained in that line of code and ask whether it is *equal to* that set value. In the case of the FACT program, as long as the value is not equal to 5, the loop must be repeated.

Clearly, the original source code did not contain, or use, a compare instruction. It simply instructed that the FOR loop shall be repeated as long as the value of *i* is less than 6. At no point did the programmer intend that the computer should check whether the number of times the loop has been completed is *equal* to 5. The programmer did not have to dictate this in the source code. He could simply rely on the rules of the programming language and the compilation process, to create the necessary instruction that would terminate the FOR loop at the correct point.

Thus, the external influences listed above are responsible for the particular expression of the instruction in this case, and not the programmer. Therefore, it is clearly improper to suggest that compilation translated the source code because, evidently, it created new code based on factors outside the message conveyed by the source code.

In addition, depending on the compiler itself and the pre-sets imposed upon it by the programmer and the software environment, the compiler makes certain additions and alterations to the object code “by adjusting and manipulating code generation in certain ways”[317] in a process known as machine-independent optimization.

As the name suggest, these steps are aimed at improving the ability of the object code to perform its intended function better. These changes are included in the object code regardless of the hardware or software environment in which it should operate. In

[316] Davidson (1983) *Jurimetrics* 377.

[317] Lin et al (2011) *Santa Clara High Technology Law Journal* 239.

addition, certain machine-dependent changes and additions may also be made by the compiler, based on prior knowledge of the set of instructions of another computer program with which it must operate in concert.[318] Consequently, depending on how the compilation process is carried out, it may result "in significantly different object code given a particular piece of source code."[319]

There is a danger that this fact may be confused, and equated with, the necessary changes that are made to a literary work during translation according to the inherent requirements of the language. However, unlike translation of a literary work, the changes made to the computer program during compilation do not depend on language rules but on the intended effect it may have, i.e. the efficiency of the program.

The same source code may give rise to different sets of instructions in object code depending on the combination of factors applied by the programmer during the compilation process, *without any change to the programming language*. By analogy, this would be similar to reworking a literary work, without changing the language, by restricting the process to a certain dictionary and pre-selecting certain colloquialisms intended to reflect a dialect. This is not the making of a translation. At best, it is an adaptation.

Therefore, the process of compilation is a translation only insofar as it is a conversion of the meaning expressed by the source code into a new set of instructions. But the process of compilation is not a neutral process. It changes the message where necessary and creates a new set of instructions, influenced by a variety of factors which are not dependent on, or related to, the original source code.

Thus, the process of compilation is as much a transformative process as it is mechanical, and the result does not give expression to human intellectual endeavour analogous to the practice of translating a literary work. In other words, while the

[318] See Lin et al (2011) *Santa Clara High Technology Law Journal* 239 for a description of the technical adjustments made during compilation for the purpose of both machine-independent and machine-independent optimization.

[319] Lin et al (2011) *Santa Clara High Technology Law Journal* 239.

programmer may make certain structural or technical decisions when drafting the source code which he intends should be reflected in the final object code, he is not responsible, directly or indirectly, for drafting those instructions in the object code that give effect to his idea or intention.

This means that compilation is neither the making of a translation nor a process akin to translation. It is a *sui generis* process.

Consequently, if compilation is not translation, it must follow that decompilation cannot be translation in reverse. There is no way for the person carrying out decompilation to ensure that all of the amendments to the object code, which were made during compilation, could be reversed to deliver the original source code. In addition, without knowing the message which the original programmer intended to convey, the decompiler must make certain presumptions about the external influences and the programming techniques that influenced the making of the original object code and, based on this information, create a set of instructions that could relay the same meaning as the original source code. Thus, all that decompilation achieves is an approximation of what the original *message* could have been and, based on that, create a set of instructions that convey a similar *meaning*.

3 2 1 1 4 Code as symbolic communication

An additional factor, which distinguishes compilation and decompilation from translation, is the fact that computer code conveys meaning symbolically, not literally or textually. In other words, the code, in whatever form or language, carries as much of its meaning in the *exact characters* as it does in the composition of words and phrases.

In the case of literary work, a mistake in the composition of characters will result in spelling or grammatical errors but the meaning of the word or the sentence will remain the same. In the case of computer programs, even a slight variation to the composition of some characters, may result in no meaning being conveyed at all. To ensure this does not happen, in most cases the compilation process does not flow directly from source code to object code and is, depending on the intention of the programmer, interrupted by any number of automated re-drafting or interpretation stages.[320]

In this respect, reference should be had to the meaning of “software engineering” as attributed to the process of compilation,[321] which means that object code and source code are, or may be considered to be, written simultaneously (the former indirectly and the latter directly). Thus, both may qualify for copyright protection because each is a distinct “encryption”[322] of instructions, albeit two different sets of instructions. In this context, encryption is the most appropriate word to describe the fact that code conveys meaning symbolically, rather than literally.

For that reason, in practice, the process of compilation is not referred to as translation but *transformation*[323] – a term which accurately reflects the engineered creation of a symbolic work by intermediate means, influenced by computer-aided and automated manipulation, subtraction, compression and addition of instructions.

---

[320] Davidson (1983) *Jurimetrics* 372, 377. See also S Karjiker 2013 *Open-source software and the rationale for copyright protection of computer programs* Thesis LLD University of Stellenbosch (2013) 155-6.

[321] Davidson (1983) *Jurimetrics* 342-3.

[322] 382.

[323] See for example M L Van De Vanter “Preserving the Documentary Structure of Source Code in Language-based Transformation Tools” (2001) *IEEE International Workshop on Source Code Analysis and Manipulation (SCAM 2001), 10 November 2001, Florence, Italy* where the author states, in relation to the nature of decompilation programs, that “[t]he defining characteristic of such tools is that they must generate transformed source code suitable for further use by people, to whom documentary structure is essential.” The same work contains a useful number of illustrations of code to support this fact. Regarding the transformative nature of compilation and decompilation, see D Waddington, G & B Yao “High-Fidelity C/C++ Code Transformation” (2005) 141 *Electronic Notes in Theoretical Computer Science* 35 at 36, where the authors state that “the process of forming an abstract representation of the program code often leads to style disruption; program layout (whitespace), commenting and use of preprocessing are not precisely retained throughout the transformation process (hence abstract).” For technical detail in an accessible form regarding the programming process and the fact that compilation is itself a process of interpretation rather than translation, see University of Hawaii System “ICS 1111: The Programming Process” <http://www2.hawaii.edu/~takebaya/ics111/process_of_programming/process_of_programming.html> (accessed July 2020). See further the sources in the footnote immediately below.

In light of the four reasons provided above, it is clear that neither compilation nor decompilation is akin to translation and, insofar as copyright law considers it a translation, it is inaccurate.[324] Furthermore, it must be understood that the term translation was used in the development of copyright law in relation to computer programs for one purpose only, namely, to provide protection for the object code along with the source code. This relied, as shown above, on a literary-analogy. While this work has shown that object code is not a translation of the source code, it is not suggested that object code is, consequently, unprotected. There is no reason to stretch the above findings to this point. All that is suggested here is that, when the legality of decompilation is determined, the literary-analogy may not be allowed to influence the decision. In other words, it is unsound to tolerate the application of the meaning of translation, which is technically incorrect, beyond the point where it is absolutely necessary. To do so would be to perpetuate the literary analogy. It is submitted that the meaning of translation, incorrect as it may be, is useful to protect object code along with source code *as a type of copyrightable work*. However, it is not acceptable to use the meaning of translation as a means to determine the nature of adaptation of a computer program because, in fact, neither compilation or decompilation amounts to the making of a translation.

[324] From a technical perspective, see for example DA Plaisted "Source-to-Source Translation and Software Engineering" (2013) 6 *Journal of Software Engineering and Applications* 30 where the author explains the range of problems in software development that flow from the fact that translation is not technically possible, which creates "the need to rewrite programs over and over again for different languages and machines. Such rewriting would not be necessary if it were possible to translate programs, or portions of programs, from one high-level language to another so that they would not have to be written from scratch in each language." See further M Van De Vanter, L "Preserving the Documentary Structure of Source Code in Language-based Transformation Tools" (2001) *IEEE International Workshop on Source Code Analysis and Manipulation (SCAM 2001), 10 November 2001, Florence, Italy* where the author states that code cannot be translated but is transformed because "the documentary structure of code (its human meaning) is grounded in information that *cannot be derived from* its linguistic structure, and in fact *cannot even be understood* in those terms" (original emphasis). In this respect, the author deals with the obstacle posed by decompilation tools which, universally, fail to reveal the original source code. As the author points out "[t]he challenge for the class of tools identified here is to construct, along with modified code, a new documentary structure that conveys the same meaning to the human reader as did the original."

Nevertheless, SA copyright jurisprudence suggests that compilation is a translation and, therefore, decompilation is an adaptation by means of reverse translation. The basis of this view is analysed below in order to conclude the discussion on adaptation by translation.

3 2 1 1 5 Adaptation by translation in SA copyright law

In SA copyright jurisprudence, the prevailing literary view of computer programs has created the impression that the act of decompilation amounts to the making of an unauthorised adaptation of the source code by means of translation, by virtue of the fact that object code is protected as a translation of the source code in terms of the Copyright Act.

As shown above, this is not correct. Any attempt to classify object code as a "cousin"[325] of source code, in order to leverage the textual view of copyright, is misplaced and both legally and technically incorrect. There is no authority in South African case law regarding the prohibition on the decompilation of computer programs. And yet, the bulk of leading academic opinion on this point, assumes that decompilation is prohibited[326] based on a reading of the Act. This assumption is based, in most cases, on the perception that object code is a translation of source code and, consequently, decompilation is a prohibited act because it amounts to the making of a reverse translation.[327] This contention has been dismissed in the analysis above based on the *technical* nature of compilation and decompilation. However, it is also necessary to question the *legal* basis of this contention, in SA copyright jurisprudence.

---

[325] Davidson (1983) *Jurimetrics* 380, See also, Soobert *John Marshall Law Review* 120-2 regarding the reliance on the exclusive rights of making derivative works and modifying (or making adaptations of) the work.

[326] Van der Merwe D P, et al. *Information and Communications Technology Law* 2ed (2016) 291; Jooste C and Karjiker S "Intellectual Property Law in the Digital Environment (EIP Law)" 414-5; Dean O H *Handbook of South African Copyright Law* 14ed (2012) 1-74 where the author does not distinguish the protection afforded to object code as an adaptation of the source code from the making of a translation of the work when this process is reversed.

[327] In some cases, the argument is also based on the perception that even if decompilation does not amount to translation, it is nevertheless a form of adaptation which involved a degree of technical reproduction and manipulation of the code. This contention is analysed in detail further below.

At the outset of this analysis, the basis of the above contention encounters a problem, because the only South African judgment to support the supposition that object code is a translation of source code,[328] and vice versa, is no longer precedent.[329] The court in *Rosy* held that "the object codes are adaptations and translations of the original source codes"[330] because the process of compilation fits within the "wider meaning being given to the word translation"[331] by a Federal court decision in Australia.[332]

The South African court undertook no analysis to arrive at this conclusion and ignored the dissenting opinion of one judge in the Australian case for no reason other than preference.[333] Furthermore, this decision was overturned on appeal by the High Court of Australia which rejected the earlier courts' wider "metaphorical" construction[334] of the word translation and held that object code *is not a translation* of the source code.[335]

---

[328] *Apple Computer v Rosy t/a SA Commodity Brokers (Pty) Ltd and Another* 1984 JOC (13) 134 (D) (*Rosy*).

[329] The reasons why the decision in *Rosy* has no persuasive value, are discussed immediately below. Other post 1992 cases, where the translation of object code to source code is mentioned, assume the accuracy of the translation-basis but provide no reason for this finding, other than the fact that both object and source code are expressions in writing. See for example *Haupt t/a Soft Copy v Brewers Marketing Intelligence (Pty) Ltd and Others* 2004 BIP 207 (C) at 219. In *Lacfin (Pty) Ltd v Johannes Nicholaas Le Roux and Four Others* 2000 BIP 190 (O) at 202 and 205, and *Econostat (Pty) Ltd v Lambrecht and Another* 1983 89 JOC (W) at 112, both courts dismissed the claims for copyright infringement because the source code was not submitted as evidence and, therefore, the court could not make a determination on substantial similarity. These cases provide no clarity on the meaning of translation but suggest, incidentally, that the source code is the primary work and the object code a translation thereof because, without a comparison of the *source codes* of both works, the court could not make a finding on copyright infringement.

[330] *Apple Computer v Rosy t/a SA Commodity Brokers (Pty) Ltd* 135.

[331] *Apple Computer v Rosy t/a SA Commodity Brokers (Pty) Ltd* 136 (original quotation marks omitted).

[332] *Apple Computer Inc. and Another v Computer Edge Pty. Ltd. and Suss* [1984] F.S.R. 481, an appeal to the single judge decision in *Apple Computer Inc. and Another v Computer Edge Pty. Ltd. and Suss* [1984] F.S.R. 246, which was later appealed to the High Court in *Computer Edge Pty. Limited and Another v Apple Computer Inc. and Another* [1986] F.S.R. 537.

[333] *Apple Computer v Rosy t/a SA Commodity Brokers (Pty) Ltd* at 136.

[334] *Computer Edge Pty. Limited v Apple Computer Inc* at 565.

[335] The court held, at 548, that "when the definition of 'adaptation' […] is read as a whole, it suggests that the primary and not the transferred and figurative, meaning of 'translation' is intended" and that the "definition of 'adaptation,' […] suggest that an adaptation must itself be a 'work.'" For these reasons,

In the following statement, the court refers to a ROM chip to mean *the object code* recorded on a memory device in this case:

> "The ROMs did not in any way express or render the source programs; rather, the ROMs were the means of putting into action and making effective the instructions written in the source programs. Just as a person does not (except in a metaphorical sense) translate the instructions for the working of a machine when, following those instructions, he sets the machine in motion, so the electrical charges in the ROMs effectuate, but do not translate, the instructions in the source program."[336]

An analysis of the merits of the Australian decision is unnecessary in this work. Although the court correctly held that object code does "not resemble the source programs,"[337] all of the decisions in this matter relied on legal interpretive findings[338] that are no longer sound. Thus, the *Rosy* judgment cannot be precedent for the contention that compilation is a translation, and certainly not for applying this idea to decompilation. The decision has no merit on its own and relies entirely on foreign law which is itself tainted by legal inaccuracies and was, subsequently, overturned. Reliance on *Rosy,* to suggest that object code and source code are protectable as translations or adaptations, is to rely on spurious authority. At the very least, *Rosy* is not dispositive of this question.

Furthermore, this finding predates the 1992 Amendment Act and dealt with computer programs as literary works, which are subject to the restriction on translations in the Act. Therefore, it cannot be cited as authority for the supposition that compilation or decompilation of computer programs created after 1992 amounts to an adaptation in the form of translation.[339]

---

the court found that the act of reproducing the ROMs, containing object code, did not amount to infringement.

[336] *Computer Edge Pty. Limited v Apple Computer Inc* at 548.

[337] At 549.

[338] That in order to be a computer program the work must be recorded in writing in the traditional sense and not just recorded on an electronic memory device, and, that to qualify as an adaptation, the infringing work must itself constitute a copyright work. See *Computer Edge Pty. Limited v Apple Computer Inc* at 543-9.

[339] The question whether or not it may amount to making an adaptation by a means other than translation, is discussed further below.

In fact, it is submitted that the judgment in *Rosy* has no residual application at all concerning the legal interpretation of the relationship between source and object code. It is irrelevant that the works in *Rosy* were, in fact, computer programs – the court dealt with the programs as literary works. The departure from this position in 1992 expressed a clear intention that the compilation of computer programs shall no longer be treated as translations. The 1992 Amendment chose to define the act of adaptation in relation to computer programs *without* including the term translation. Thus, the position in *Rosy* is no longer compatible with the type of work or the meaning of adaptation.

The only other SA jurisprudence on the adaptation of computer code is the *Marconi*[340] decision, where the court held that the act of removing portions of code amounted to the making of an adaptation of the program, despite the fact that the alteration of code is *not a listed form of adaptation* in the act.[341] The court based its finding of adaptation on the prohibition to amend the program, contained in the EULA (the click-wrap license), which prohibited adaptation *per se*. Thus, the court found infringement because "there is no requirement that the person who does an unlicensed act must ensure that, in doing so, he or she infringes copyright."[342] This decision has been correctly dismissed as precedent on the meaning of adaptation in relation to computer programs.[343] Therefore, it can hardly be precedent for the contention that the meaning of adaptation may be extended even further to include translation, even if decompilation could be considered a translation.

---

[340] *Technical Information Systems (Pty) Ltd v Marconi Communications (Pty) Ltd* 2007 1047 JOC (W) (*Marconi*).

[341] Adaptation in relation to a computer program may take one of the following forms, but is not a closed list:

> "(i) a version of the program in a programming language, code or notation different from that of the program; or
>
> (ii) a fixation of the program in or on a medium different from the medium of fixation of the program."

[342] *Technical Information Systems v Marconi Communications* at 32.

[343] Dean O H *Handbook of South African Copyright Law* 1-74 para 8.5.9 where the author submits that the act of removing portions of code "does not fall within the ambit of what is contemplated by the restricted act of 'adaptation' in the Copyright Act."

Furthermore, from a normative perspective, there is local precedent which suggests that the literary meaning of translation is inappropriate in the case of computer programs. The court in *Econostat* dealt with the contention that, by using the "raw data"[344] in a database and creating a computer program to process that data in a manner that is similar to the plaintiff's computer program, the respondent infringed copyright in the data and the computer program, as a literary work. The court dismissed this contention because the plaintiff failed to prove ownership of copyright in the data or the computer programs.[345] Despite this finding, the court made several obiter remarks about the plaintiff's likelihood of succeeding with an infringement claim in the event that it could prove ownership. In this respect, the court held that the plaintiff would likely not succeed with a claim that the act of re-writing and reorganising[346] the computer program in a different language, in order to be compatible with a different type of computer, amounts to infringement.

The court referred to English scholarly opinion and US case law to support its opinion that the re-use of the "idea, concept or brainchild"[347] of the programmer, to create a new program in a different language, does not constitute infringement.[348] The court opined that, unless there is substantial copying of literal code, a case of infringement has not been made.[349] Thus, the act of translation alone would not be sufficient to support a contention of infringement. However, these remarks were obiter and are, therefore, not dispositive on the nature of adaptation by translation. The judgment may, at best, be indicative of the court's discomfort in applying translation as a form of

[344] *Econostat v Lambrecht* at 104. The data was "extracted from government or other published documents, supplying economic information, prices, statistics, etc.". See *Econostat v Lambrecht* at 105.

[345] *Econostat v Lambrecht* at 111.

[346] At 97.

[347] At 111.

[348] It must also be acknowledged that, in terms of section 2(3) of the Copyright Act, it is possible to vest copyright in a work even though that work infringes the copyright of another work. In such cases, the protection in that work would extend only to those parts that meet the originality standard, i.e. those parts that are original to the author and not the result of infringing reproduction or adaptation. See further Dean O H *Handbook of South African Copyright Law* 1-24 para 3.3.1 and fn 9. Thus, whether or not decompilation is a form of infringement, the program created after decompilation took place is nevertheless protectable.

[349] At 112.

infringement in the case of computer programs, where no evidence of literal reproduction has been submitted.

The court in *Econostat* also dismissed the possibility that infringement, by adaptation, might be established on the basis that the program was derived from the computing method or system expressed by the original program.[350] This argument is addressed further below in this chapter.[351]

In the absence of any other reason to suggest that decompilation amounts to the making of an adaptation by translation, it would be a mistake, brought about by the overbroad application of the literary-work analogy, to argue that decompilation of computer programs, in SA copyright law, amounts to the making of a translation. Not only is this technically incorrect, there is no legal basis for this assumption.

However, while the act of decompilation does not translate the object code, the technical reality is that decompilation delivers a set of instructions which might convey the message of the original source code to some degree. In other words, inasmuch as the process of decompilation creates a new set of instructions, the contents thereof will convey, to some extent, the meaning which the original creator intended to express in code. To access and read this is, after all, the primary purpose of decompilation. In other words, there is a clear association between the source and object code and, by decompiling the object code, the work is manipulated in order to discover the underlying meaning.

This raises the question whether decompilation may be prohibited on the basis that it amounts to the making of an unauthorised adaptation of the work by some other means. This requires a closer inspection of the provisions of the Act. In relation to computer programs, an adaptation includes a version of the work in a different "programming language, code or notation."[352]

---

[350] *Econostat v Lambrecht* at 112.

[351] See paragraph 3 2 2 1 3 below.

[352] The Copyright Act section 1(1) definition of 'adaptation' in relation to a computer program.

Having dismissed the possibility of considering decompilation as a form of adaptation by translation, the enquiry now turns to the other forms of adaptation.

3 2 1 2 Adaptation by means of a change in language

It has already been shown that source and object code are not expressions in different programming languages but different, increasingly lower, abstractions of the original instructions. Only the source code is written in a programming language. In the above illustration, the source code was written in the C programming language. Thereafter, during compilation, the assembly and object code is created, which are abstractions of the instructions in generic computer lexicon, assembly or binary code form, neither of which represent a programming language. As explained above, the creation of the assembly code and the object code is influenced by external factors, one of which is the programming language. Thus, the additions or alterations made by the computer during compilation, relies on the fact that the programming language is not changed.

The fact that object code is sometimes referred to as machine *language* does not mean it is a different programming language. Machine language is a synonym for machine-readable code. The same applies to assembly language, which is not a programming language but a level of abstraction of computer code, expressed in generic terminology. This is the reason why the result of decompilation is usually provided in assembly code, or assembly language, because it is universal. It is impossible to determine, based only on the object code, which programming language was used to write the original source code. Thus, the decompilation process does not attempt to deliver a result in source code in a particular programming language but, instead, delivers a generic set of instructions, in assembly code, which could have been created by a set of instructions written in any programming language.

As the source code is transformed into object code during compilation, the programming language is not changed. In short, programming language refers to the expression regarding the source code only. Thus, adaptation by changing the programming language only occurs when the original source code is redrafted in order to suit the rules of another programming language. At no point during compilation or decompilation will a change in programming language occur.

Thus, the question of language does not enter the current debate on adaptation and, consequently, the first part of the definition of adaptation, namely "programming language," does not prohibit the decompilation of a computer program.[353]

The second part of the definition, namely "code or notation", requires more extensive consideration.

3 2 1 3 Adaptation by means of a change in code

It has been shown that the process of decompilation creates a new set of instructions in assembly code or source code form,[354] derived indirectly from a work in object code form. In other words, there has been an apparent change in 'code', provided that it can be said that the new code is an adaptation of the original because it *reproduced* the original in a different code.[355]

The question is, therefore, whether this process amounts to the making of an adaptation *per se*, insofar as the process is potentially prohibited by the Act. The reason for this enquiry appears to be obvious – the process of compilation takes something that is in source *code* and creates something in object *code.* The reverse process uses a work in object code to create a work in source code which, it follows, must therefore be a prohibited adaptation.

However, this assumption is incorrect. It has been shown that the distinction between source code and object code is an artificial separation which seeks to fix a threshold, or moment of abstraction, in the programming process at which point the humanly legible instructions are converted to illegible machine-readable object code expressed

---

[353] Even if it could be argued that decompilation is a form of translation which delivers the same original source code in a different programming language, it would nevertheless fail to be an adaptation for the same reasons discussed below in relation to code and notation.

[354] As shown above, the assembly code created during compilation is both object code and source code.

[355] For the purpose of posing this question, one must bear in mind that copying is an element of adaptation. This point is discussed further below regarding the reproduction of a reproduction of a protected work.

as binary code.[356] In reality, this is a gross oversimplification and a conflagration of distinct technical measures.

While it may be true, in general terms, that source code and object code represent the highest and lowest levels of instructing the operation of the computer, they are not distinct *codes*. In fact, source code and object code, created by the original programmer, have in common their symbolic nature as a means of conveying an instruction in a mathematically precise manner and is, therefore, a codification of the same meaning and intent. This point should not be confused with translation. The source and object code are distinctly different expressions and are not translations of the message. However, the source and object code are both codifications *per se* – an expression that conveys the same meaning in an abbreviated, symbolic manner. In other words, the fact that the word 'code' is used to distinguish between the two outermost points of abstraction, namely source and object, should not be misunderstood to imply that these are different codes. The technical nature of compilation and codification has been discussed above, in relation to code as symbolic communication,[357] where it was made clear that the source and object code are different encryptions, i.e. more or less complex abstractions, which are created based on the rules of a single code.[358]

The fact that there are material differences in legibility or notation between the instructions drafted by the programmer and the instructions executed by the computer is irrelevant for the purpose of identifying whether or not decompilation amounts to the

---

[356] Some of the most frequently citied academic works on copyright in computer programs repeat, with little or no variation, the same sentiment. See for example: Kravetz P I "Copyright Protection of Computer Programs" (1998) 80 *Journal of the Patent and Trademark Office Society* 41 46; Spivack P G "Does form follow function? The idea/expression dichotomy in copyright protection of computer software" (1988) 35 *UCLA Law Review* 723 730-1; Miyashita *The John Marshall Journal of Information Technology and Privacy Law* 45; Gesmer L T "Developments In The Law Of Computer Software Copyright Infringement" (1986) 26 *Jurimetrics* 224 n2; Martin A C and Deasy K "Licensing of Intellectual Property Rights Needed for Software Support: A Life Cycle Approach" (1988) 28 *Jurimetrics* 223 230; Stern R H "Another Look At Copyright Protection of Software: Did the 1980 Act do anything for object code" (1981) 3 *Computer Law Journal* 1 2-4.

[357] See paragraph 3 2 1 1 4 above.

[358] See the technical sources discussed in fn 323 and 324 above.

making of an adaptation by a change in code. The source code and object code are the same insofar as both are codified, symbolic shorthand, instructions conveying the same general meaning.

The inevitable conclusion is that source code and object code are not different codes, but different levels of complexity of a coded message – it expresses the same intended mathematical calculations, in different ways and in different terms, but using the same rules of a particular code. These rules are determined by, among others, the programming language, the specific compiler program and external requirements dictated by the environment in which the program is intended to operate.

In other words, the source and object code forms represent the two poles of more or less complex and abbreviated encoding. Therefore, to consider compilation as a change in code, rather than merely a greater compression of the same coded message, is incorrect. Consequently, there is no change in code when the reverse process is carried out.

Decompilation does not change, and therefore does not adapt, the code. It merely interprets the code and creates a set of instructions at a higher level of abstraction. In the same way that compilation processes the initial instructions at increasing levels of complexity, the decompilation process attempts to decrease the level of complexity in the object code to a more accessible representation. In other words, when a program is compiled it increases the level of codification, while decompilation decreases it and estimates what the original coded message might have been. But, in both cases, the code remains the same, only the complexity thereof changes. If compilation changed the rules of the code to deliver a different codification, the program would not work as the programmer intended.

At this point in the analysis, it has been shown that decompilation is not the reverse of compilation, does not deliver the original source code or translate the object code. It has also been shown that decompilation does not convert one code to another or change the language. Thus, all but one form of adaptation has been dismissed as a possible basis to consider the act of decompilation a form of infringement.

The last option, namely, a change in notation, requires a different approach, insofar as it does not rely on a technical interpretation of decompilation but, instead, on a legal analysis of the meaning of adaptation in copyright law.

The rigorous separation between source code and object code into distinct forms of expression, and distinct codes, has been adopted in copyright jurisprudence *ab initio*. The reason for this is clear and is based on the literary-analogy. Unless source code and object code are treated as directly related translations of the same work, the traditional approach to infringement by adaptation of literary works would not be able to address the postulated misapplication of object code.[359] This is the, admittedly precarious, basis for the argument that decompilation must somehow be an adaptation because the traditional sentiment that "taking the heart of the original and making it the heart of a new work was to purloin a substantial portion of the essence of the original"[360] must also be applicable to computer programs, particularly since it is analogous to literary works.

It is, thus, not surprising that the mistaken assumption, outlined above, has been perpetuated in South African copyright law.[361]

3 2 1 4 Adaptation by means of a change in notation

It is clear that the process of compilation changes the extensive English language "overlay",[362] or source code, into a shortened, symbolic record of the meaning of the original instruction, with additional changes and amendments. Thus, despite the fact that both rely on the same underlying code, and the rules associated with interpreting that code, the initial instructions and the assembled or compiled instructions will consist of different texts which have little, if any, resemblance as literary text and yet convey, in broad terms, the same meaning.

---

[359] Bainbridge D *Legal Protection of Computer Software* 5ed (2008) 268.

[360] *Acuff-Rose Music Inc v Campbell* 972 F.2d 1429 at 1438.

[361] See for example: Staines *Modern Law Review* 233; Pistorius et al *SA Merc LJ* 351; Van Der Merwe D et al *Information and Communications Technology Law* 2ed (2008) 291; Dean O H *Handbook of South African Copyright Law* 1-19 at 2.12.5; Jooste C and Karjiker S "Intellectual Property Law in the Digital Environment (EIP Law)" 414-5.

[362] Davidson *Jurimetrics* 341.

This is also true for decompilation, insofar as the process delivers a set of instructions, in a form of writing that is different to the object code but conveys the same meaning. This means that there is, in principle, the possibility that, decompilation may be considered an adaptation by a change in notation, of the object code.

In fact, this will usually be the case because decompilation aims to make that which is difficult or impossible to read, or interpret, legible and more accessible. Therefore, decompilation does amount to an adaptation of the *object code*. This does not, however, mean it is a prohibited act. This work has not found any authority in case law which considered the meaning of notation in the context of computer programs. Nevertheless, for the purpose of this analysis, it is accepted that the meaning of the term notation is wide enough to encompass any change in the writing, or literary expression, that is consistently different to the original.

However, in order to be an infringing adaptation, it must also be shown that the decompilation result, which is a change in notation, was created by *copying* the object code because, as discussed below, copying is an inherent element of the restricted act of adaptation.

For the purpose of, temporarily, concluding the current analysis, it is submitted that decompilation does not amount to adaptation by a change of notation because the process of decompilation does not involve copying of any part of the object code into the decompilation result. The reason for this finding will become clear in the course of the following discussion, which deals with copying and the restricted act of reproduction. Insofar as the analysis below concludes that the element of copying is not satisfied during decompilation, it is applicable to all of the forms of adaptation discussed above.

Therefore, if the above conclusions, regarding the change in code or programming language, are considered to be insufficient to substantiate the finding that decompilation is not an adaptation, the finding below, regarding the element of copying, supports the accuracy of the findings made above.

3 2 2 The restricted act of reproduction

Four contentions were listed above which might support the argument that decompilation is an infringing act. The first two, which are based on the restricted act of adaptation, have been discussed above.

The remaining two contentions, which are based on the restricted act of reproduction, are: (1) that decompilation reproduces the original source code because it copied the object code, a protected reproduction of a reproduction, into the decompilation result or, alternatively, reproduces the object code into the decompilation result; and, (2) that decompilation reproduces the protected object code by temporary or intermediate copying as part of the technical process.

It must be noted that the first contention is based on the restricted act of reproduction, but it also relates to the restricted act of adaptation. Thus, the contention is that decompilation is a form of adaptation because there has been a degree of copying. Insofar as copying is the basis of either reproduction or adaptation, the contention is discussed below.

Conversely, the second contention, discussed further below, is based only on the restricted act of reproduction and argues that decompilation involves the making of infringing reproductions in the form of intermediate copies.

3 2 2 1 Reproduction of the code into the decompilation result

If the process of compilation is not translation, decompilation cannot be re-translation. From this, it follows that the process of decompilation does not, in fact, amount to the making of a reproduction *of the source code*.[363]

As shown above, decompilation is, at best, described as a process to arrive at a *simulation* of the source code. Thus, at the outset of this analysis, a misalignment has been identified between the legal perception of decompilation and the technical reality. However, copyright law protects both source code and object code by, inter alia, the

[363] The possibility does exist that decompilation is a reproduction of the *object code,* by way of intermediate copying. This is discussed further below in paragraph 3 2 2 2.

application of the principle that a reproduction of a reproduction of a work infringes copyright in the original work. Therefore, although it has been shown to be technically incorrect, the object code is legally protected as a reproduction of the source code. Consequently, decompilation may, in theory, be considered a reproduction of the source code, provided that it is shown that decompilation involved copying of the object code. This point is discussed further below. First, it is necessary to discuss the degree to which copying is an essential element of adaptation.

Generally, in order to succeed with a copyright infringement claim by means of adaptation it must be shown that there was "actual copying"[364] of the protected work. The extent to which copying must be evident will depend on the facts. However, it is clear that, for the purpose of adaptation, the degree of similarity need not be substantial to the same degree as would be expected in a case of infringement by reproduction.[365] If the same level of copying is required, all instances of adaptation would be actionable as cases of infringement by reproduction, rendering the exclusive act of adaptation useless.

---

[364] Dean O H *Handbook of South African Copyright Law* 1-75 at 8.6.3 states "[it] is essential to appreciate that copyright in a work is only infringed by the unauthorised reproduction of it, or the unauthorised adaptation of it, if there is actual copying." It appears that the author used the word "copying" in this context, rather than 'reproduction', to suggest that in the case of adaptation, the degree of literal reproduction may be lower, or the extent of copying less obvious, than a case of infringement based on reproduction. The 'copying' element of adaptation is not merely a test of whether there is a causal connection between the original and the allegedly infringing works, it is a part of the nature of the restricted act. In order to prove infringement by adaptation, it must be shown that actual copying took place, even though it may not be literal or evident in a textual comparison. See Van Caenegem W *Intellectual Property Law and Innovation* 122. In the UK, this is sometimes referred to as indirect copying. See Bainbridge *Intellectual Property* 256. In South Africa, the essential requirement of copying is clearly established in relation to infringement by reproduction or adaptation. See for example *Galago Publishers (Pty) Ltd and Another v Erasmus* 1989 (1) SA 276 (A) at 281C. See also, at 290D and 291E-G, where the court deals with the "pattern of copying" which established an "impressive case of copying". See further *Dexion Europe Ltd v Universal Storage Systems (Pty) Ltd* 2003 (1) SA 31 (SCA) at 37[4]A-B where the court makes it clear that copying is a requirement and part of both steps of the test for infringement by means of reproduction or adaptation.

[365] In fact, it is clear that in some cases the meaning of 'copying' is given a wider meaning than the term 'reproduction'. See MacQueen, et al. *Contemporary Intellectual Property Law and Policy* 138. See also the analysis in paragraph 3 2 2 1 3 below.

The restricted act of adaptation is intended to address those instances where the infringing work contains a variation of the original work, or can be said to be derived from it.[366] Thus, the original work was substantially altered or transposed in such a way that the act of copying is obscured. Nevertheless, the infringing work must illustrate that copying of the original took place. Thus, although the work need not show a verbatim reproduction of the disputed parts, the similarity between the works must be due to an act of copying followed by an attempt at manipulation.

However, in the case of decompilation, copying of the object code occurs only at one point when the object code is loaded into the computer memory in order to be processed by the decompilation program. The question whether or not this amounts to infringement is discussed further below in relation to the fourth contention, namely, reproduction by intermediate copying. For present purposes, it suffices to point out that, except for intermediate copying, decompilation does not involve copying from the object code into the decompilation result. As shown above in the illustration of the FACT program, there is clearly no textual correlation between figure 3, the object code, and figure 4, the decompilation result. Thus, the argument that decompilation involves copying of code must rely on a wider meaning of ‘copy’.

This wider meaning, for the purpose of adaptation, is usually based on the fact that there is a substantial similarity between the two works which, although not literally identical, is so widespread, or represent the unique expressions of the original work, that it could not have been arrived at by a process of independent creation and must, therefore, be the result of copying.[367]

In the case of decompilation, it is submitted that this process will deliver a false positive finding of copying, for the following reasons.

[366] See Dean O H *Handbook of South African Copyright Law* 1-73 para 8.5.3 where the author suggests that “the concept of transforming a work is evident in the definition of ‘adaptation’ in the Act.”

[367] See the case discussions in chapter 4 below, where this principle is illustrated. See in particular paragraphs 4 2 3 5 and 4 2 3 7.

3 2 2 1 1 Copying occurs in isolation

For purpose of the current analysis, it is submitted that any similarity of code that may be evident between the source code and the results of the decompilation process, will be as a result of the estimation process carried out by the decompilation program and *not* as a result of copying object code instructions. In other words, the similarities are due to the process of reading the object code and writing new source code that may perform the same function. It is, therefore, an independently created set of instructions. This process is not analogous to adaptation in the traditional sense, where a part of the work was copied to deliver a derivative version. In the case of decompilation, the act of copying is isolated from the decompilation result – it occurs only once, automatically, when the object code is loaded into the machine.

Consequently, there is no copying of the object code into the decompilation result. In fact, there is no copying at all outside the internal operation of the computer. The decompilation result does not contain a copy of the object code because the decompilation result sought to create a set of instructions that illustrate the *message* conveyed by the object code. The purpose of decompilation is to create a legible set of instructions. Thus, there is no point in copying the object code into the decompilation result because it will remain illegible.

This means that, because there is no reproduction of the object code into the decompilation result, it cannot be said that the similarities between the decompilation result and the original source code are due to reproduction. As shown above, the similarities are only due to the sophistication of the decompilation process and the degree to which it accurately estimated what the original source code could have been.

3 2 2 1 2 The technicalities exclusion

Furthermore, in an adaptation case it should not be argued that the degree of similarity between the decompilation result and the original source code is indicative of copying – the work must be reviewed in light of all of the circumstances under which it was created before a causal connection is established.[368] In the case of computer

[368] That is why it is essential that the circumstances under which the work was made, strongly suggest that the work could not have been created by, or the similarities are not the result of, a process of

programs, this fact is important. Despite the wide variety of programming languages, and relative freedom it affords to the programmer, all computer programs must ultimately conform to a certain standard in order to operate the machine. The standard is, as shown above, variable to some degree but share certain common features, rules and limitations which, inter alia, require that some instructions must contain, or be expressed in, specific words or symbols or comply to the inherent rules of logic and mathematics.

Thus, any two independently created computer programs designed to operate in the same environment, regardless of their function, will share certain instructions which appear to be duplicates. If the two programs are designed to perform complimentary or similar functions, the degree of similarity will increase as a result of the need to conform to certain expected standards.

If these commonalities are allowed to create an impression of copying, it would allow copyright to vest in a programming language itself or the base expressions of computer programming. It would be tantamount to a finding that the frequency with which the words “if” and “then” appear in a book is an indication that the work was copied from another work in which these words appear. The causal connection analysis should thus consider, as part of the circumstantial factors, the technical peculiarities of the work and the mandatory expressions incumbent on the author.

Where any of the apparently copied text is a consequence of the programming process, rather than the intentional reduction to material form of the programmer, it must be ignored regardless of how peculiar each instruction, or the overall impression created by the volume of similarities of this kind, may appear in the particular case. This argument has been at the heart of much foreign case law on copyright

---

individual creation. See Dean O H *Handbook of South African Copyright Law* at 1-75 para 8.6.3 where the author states:

> “The contentious reproduction or adaptation must be a derivative of the copyright work.”

infringement in software, which gave rise to that part of the abstraction tests that attempts to filter out these unprotectable elements.[369]

In the case of a decompilation, this fact is important because the expressions in the decompilation result are a result of the operation of the computer and not a human author. The decompilation process is designed to estimate how the function of the program under review could be achieved in the specific technical circumstances dictated by the applicable settings. Therefore, the text of the decompiled work will contain the common terms, turns of phrase, logical composition and syntax, among other features, of a program that was designed to perform its function with success.

Consequently, when comparing the decompiled work with the original work, the similarities may not be allowed to create the impression of copying because they are not the result of copying but, instead, a consequence of the sophisticated approximations made by the decompilation program. In other words, where similarities exist, it will be the result of an accurate deduction and not an example of copying. Furthermore, where two sets of program instructions appear to be identical, the court has held that this is only *prima facie* evidence of copying.[370] Thus, evidence on the cause, or source, of the similarities, such as the accuracy of the decompilation process, should be considered before copying is inferred.

This means that, in order to support an argument that decompilation infringes on the restricted act of adaptation, the copying element will only be met if similar instructions

[369] Most notably *Computer Associates International, Inc. v. Altai, Inc* 1992 982 F.2d 693; *Lotus Development Corp v Borland International Inc* 1995 49 F 3d 807; *Whelan Associates v Jaslow Dental Laboratory Inc* 1985 609 F. Supp. 1307 and *Whelan Associates Inc v Jaslow Dental Laboratory Inc* 1986 797 F.2d 1222. The root of the test is traced to the "successive filtration test" suggested in the 1993 issue of Nimmer D and Nimmer M B *Nimmer on Copyright Publication 465 Release 108 August 2019* (1978). See Shemtov N *Beyond the Code* 128-9 for a critical analysis of Nimmer's test. The court in *Oracle America Inc v Google Inc* (2014) 750 F.3d 1339 rejected the construction of the test as espoused in the *Lotus* case.

[370] See *Logistics Network (Pty) Ltd v Hard & Software Systems CC and Others* 1999 BIP 278 (C) at 281. In this case, the programs were identical and the court held that in the absence of any differences between the code of the two programs, *prima facie* reproduction had been established.

are present in the decompilation result *because* it was copied from the object code. This does not mean that the instructions must be identical, but merely that it was produced by means of copying. If this element is not satisfied, the decompilation result is not an adaptation of the object code.

Considering that, as shown above, the decompilation process does not copy the object code into the decompilation result, it cannot be said that the commonalities in the pre- and post-decompiled works are indicative of copying to the extent that it satisfies the copying requirement. As shown above, the majority of similarities will relate to unprotectable, standard programming terms or techniques and not the protected expression.

Thus, it is submitted that decompilation does not amount to the making of a copy for the purpose of infringement by adaptation because: (1) no copying of the object code into the source code occurs; and (2) the similarities between the original source code and the decompilation result is not due to copying in the wide sense but, instead, attributed to common programming expressions and the degree to which the decompilation program correctly estimated that the particular line of code could have been written by the original programmer.

3 2 2 1 3 Reproduction and derivation

Thus far, it has been established that copying does not occur during decompilation. Despite the incidental similarities between the texts, the decompiled work does not contain actual copies of instructions from either the source code or the object code of the original. Decompilation merely *used* the object code as a set of instructions for the computer to deliver a result which is simply an explanation of how the program works.

During this process the original instruction is not copied and then reworked by the computer to deliver a legible result. The decompilation program read the instructions and estimated what its meaning is and, thereafter, composed an instruction that could, in the ideal circumstances envisioned by that computer, perform the same function. Only if the computer is correct on all counts will the original and the decompiled instruction be similar.

As shown above, in figure 2 and 4, the object code and the decompilation result are not identical. If the decompilation process was capable of copying and translating instructions in reverse, the similarity would be immediately apparent. Thus, insofar as copying is concerned, the process of decompilation is technically identical to the operation of the computer when a user executes a program – the code is read and interpreted and the machine delivers a result accordingly. Where that result contains the same or similar coded instructions, it is not a copy of the original but a new statement created as a consequence of the operation of the machine.

However, as mentioned above, the meaning of reproduction in copyright law is wider than the meaning of copying. Thus, in order to substantiate the argument that decompilation is not adaptation, because it does not copy code, it is not sufficient to point out that no copying has occurred. The copying element may also be met, in the absence of literal copying, by showing that the decompilation result was derived from the object code. In this context, derivation implies that the original expression was transformed into the new expression.

In SA, the courts have, on occasion, created the impression that a wider meaning than copying should be attributed to the word reproduction to encompass *derivation* from the original, without literal text borrowing. This is particularly relevant in the case of adaptation, where it is felt that the absence of direct copying should not obstruct a finding of infringement where the derivative work is too closely analogous to the original. Thus, copying is widened to encompass what may be called *inspirational use*. The basis of this view, in SA law, is reviewed briefly below.

In *Bosal v Grapnel*,[371] the court applied a peculiar meaning to the word "adaptation" and found that it should be read to mean "use"[372] of a work[373] to the extent that the subsequent work was *derived* from the original. Considering that the decompilation result is also derived from the object code, the impression is created by the judgment

[371] *Bosal Afrika (Pty) Ltd v Grapnel (Pty) Ltd* 1985 (4) SA 882 (C) (*Bosal*).

[372] At 893.

[373] At 894.

in *Bosal* that the copying element of adaptation is met when the original work is used to arrive at the set of new instructions, even though no literal copying took place.

The court applied this construction in order to read an act into the definition of "adaptation" that was not listed as an example. The fact that the list is prefaced by the word "including," made this possible. On the same basis, decompilation could be read into the definition of adaptation, based on the wide meaning of copying expounded in this case. However, in the discussion below it is submitted that the judgment in *Bosal* is not applicable to adaptation of computer programs and that the wide meaning attributed to copying in this case, should not be applicable to decompilation.

In this case, the conduct complained of amounted to the systematic calculation of a new series of spare-parts numbers, by processing an existing list of numbers according to a set formula. The act of derivation was, in this case, clear. It involved nothing more than subtracting the original six-digit parts number from the number 999999 to arrive at a new number.[374] By systematically doing so, the court found that the defendant copied *the system* that the plaintiff used to create its series of original numbers. It was clear that copying in the wide sense occurred because, as the court indicated, the calculation could be reversed to reveal the *exact* number of the plaintiff.[375] This, the court found, is a form of adaptation *of the system* that created the original number and therefore an infringement of copyright.

However, the same cannot be made applicable to decompilation. It has been shown above that the decompilation result is not a derivative version of the source code or the object code. It is an entirely new work based on the object code but also on a plethora of other technical factors. The extent to which it is *per se* derived, and therefore copied from the original in the wide sense, is not comparable. However, the possibility is acknowledged that the meaning of adaptation, extended to mean 'derived from', is capable of application to decompilation, to the extent that it might satisfy the essential copying requirement in the wide sense.

[374] *Bosal Afrika v Grapnel* at 891.

[375] In *Bosal Afrika v Grapnel* at 891 the court states that by subtracting the defendant's number from 999999 one arrives back at the plaintiff's number used to create the derivative number.

But, it is submitted that the meaning attributed to the term adaptation in *Bosal* should not be extended to computer programs because it is not, like the work in *Bosal*, a literary work. The extended meaning of adaptation was developed in relation to the provisions of the Act that apply to literary works. The act of adaptation is defined differently for computer programs. It too is capable of including other forms but, it is submitted, only where that form of adaptation has been developed in a case dealing with computer programs.

Despite the fact that the court dealt with the interpretation of the word "adaptation"[376] in general, there is no indication that it considered the possible implications of its finding on other types of work or intended to clarify the term in relation to all types of work. Furthermore, the judgment was delivered before the introduction of computer programs as a *sui generis* type of work. Thus, the meaning of adaptation in relation to computer programs should not be interpreted with reference to the judgment in *Bosal*.

Although the judgment in *Bosal* may be understood to mean that, in some cases, using a literary work to create a derivative text may amount to an adaptation, it can go no further.[377] At least not to the extent that it is authority for a contention involving adaptation by decompilation.

Furthermore, insofar as the use of an existing system to create a new derivative work might amount to adaptation according to the court in *Bosal*, local precedent suggests that this will not be the case where the work in question is a computer program, even if it is viewed, or treated, as a literary work.

In *Econostat*,[378] the court was faced with a similar contention regarding the use of a system to create a derivative work, in this case, in the form of a databank and a computer program. The court found that the alleged substantial similarity in the

[376] *Bosal Afrika v Grapnel* at 893-4.

[377] See also *Southco Inc v Kanebridge Corporation* 2004 390 F.3d 276 at 282 which held that a parts numbering system is not infringed by the use thereof because the system itself is not protected by copyright law and the individual part numbers created by the system do not meet the standard of originality.

[378] *Econostat (Pty) Ltd v Lambrecht and Another 89 JOC (W)* (*Econostat*).

numbering system and the naming method between the original work and the derivative work,[379] indicates that the defendant used the original work. However, the court opined that this does not amount to adaptation of the original work because these elements are not original[380] and, unless literal copying of protected expression is also proved, would not be sufficient to support a contention of infringement:

> "I have great difficulty in seeing how numbering as such can be considered copyright material or, in the absence of fuller explanation and details, how the use of such numbers can constitute substantial use of the copyright material. I have the same difficulty with the naming method used. There is nothing unusual or imaginative in the names and would no doubt be used by countless other people in the economic field."[381]

Thus, the opinion of the court in *Econostat* suggests that the meaning of the copying element of adaptation, in the case of computer programs, should be narrower. There are several reasons why the interpretation in *Econostat* is preferable, for the purpose of an analysis of the restricted acts in relation to computer programs, to the court's expanded reading in *Bosal*. First, the decision in *Econostat* dealt with computer programs specifically, while *Bosal* did not. Second, the judgment in *Econostat* was delivered two years before the decision in *Bosal*, but the court in *Bosal* did not refer to the *Econostat* case and, therefore, did not overturn or contradict the opinion of the court in *Econostat*. Thus, although the remarks in *Econostat* were obiter, there is no convincing reason why the wide interpretation of the copying element in *Bosal* should be applicable to computer programs while a contrary judicial opinion, which dealt specifically with this type of work, still stands.

Furthermore, in *Prism Holdings v J Liversage*[382] the court held that "there is, of course, **nothing to prevent the respondents** … from writing programs *de novo*, and selling them in competition with the applicants"[383] provided that this information is not

[379] The court dealt with data and the computer programs at the same time and did not distinguish, for the purpose of its obiter remarks on the likelihood of infringement, between the two manifestations of work. In this case, both works qualified for protection as literary work.

[380] *Econostat v Lambrecht* at 105-7.

[381] At 112.

[382] *Prism Holdings Ltd and Another v J Liversage And Others* 2001 BIP 114 (W).

[383] At 124 (emphasis added).

confidential and the programmer is not restrained from using the information in terms of the employment contract.[384] In this case, the respondents admitted to using the original code to create derivative programs. However, the court clearly stated that *nothing,* other than the contractual prohibition, prevents the creation of a derivative program, provided that literal copying did not take place.[385] This means that the court did not support a wide reading of the meaning of copy in the case of adaptation of a computer program.

Consequently, the argument put forward above, namely, that decompilation amounts to an adaptation because the object code is used by the computer and the decompilation result is derived therefrom, must be dismissed. The extent to which derivation is an adaptation of a copyright work in SA law has not been developed to the point where it is applicable to the manifestly more complex, sui generis process of decompilation.

Furthermore, local scholarly opinion on this point suggest that the term "'reproduction' should be interpreted restrictively"[386] in the case of computer programs where reliance is placed on a reproduction of a reproduction of the program,[387] such as the decompilation result.[388]

And in the UK, the prevailing view is that there is a limit to the meaning of copying and that "at a certain point, a transformation in the content of a work is so extensive that it

---

[384] At 124. The impact of a licensing restriction on decompilation is discussed further below in chapter 4 and examined critically again in chapter 5.

[385] In this case, the court was unable to determine, on the limited evidence before it, whether literal reproduction took place. See *Prism Holdings Ltd and Another v J Liversage And Others* at 123.

[386] Van der Merwe D *Computers and the Law* 2ed (2000) 43.

[387] Van der Merwe makes this point in relation to interoperability and the problem that a wide meaning of reproduction creates where several works rely on the output, or derivative works, of a computer program in order to process data or perform its function. In US and UK jurisprudence, this is analogous to interface specifications.

[388] The prevailing view in copyright law, as explained above, is that the decompilation result is a potential reproduction of a reproduction (the object code) of a protected work (the source code).

cannot be treated as a reproduction."[389] It has been shown above that decompilation is a process of extensive, repeated and complex transformation which, it is submitted, should not be treated as a copying.

This work does not examine any other SA case law[390] on the meaning of copying in the context of adaptation because it is submitted that the enquiry, about whether or not decompilation amounts to the making of an adaption, should not be pursued any further. Sufficient reason has been advanced above to make it clear that decompilation does not involve the literal copying of code and that a wider meaning of adaptation has not been developed in case law to the extent that it is applicable to computer programs. Because no SA case law exists to support the wide meaning of copying of computer programs, any attempt to find authority on this point will, inevitably, rely on a literary-analogy. As discussed above, the literary-analogy is inappropriate when the application of copyright principles to computer program is interpreted. Thus, no further discussion on the wide meaning of copying in SA copyright law is merited.

In addition to these reasons, the discussion of the wide meaning of copying for the purpose of adaptation must conclude at this point because it has no comparable basis in foreign copyright law regarding decompilation. Thus, to pursue this line of argumentation further would be conjecture. It is shown below that in both US and UK

[389] Bently L and Sherman B *Intellectual Property Law* 4ed (2009) 146. The authors' view is that the judgment in *Infopaq International v Danske Dagblades Forening* 2012 CJEU C-302/10 supports a narrow construction of the reproduction right and that this view should overrule the court's opinion in *SAS Institute Inc v World Programming Ltd* CJEU C-406/10 [2012] R.P.C. 31 [2013] Bus. L.R. 941, where a wider interpretation of reproduction was applied. It must be noted that in the *SAS Institute* case, no decompilation took place and the court dealt with the wide meaning of reproduction where a description of the program, contained in the manuals, was used. It is, therefore, not authority on the wide meaning of reproduction by means of decompilation. See further the analysis in chapter 4 below at paragraph 4 3 4 *et seq.*

[390] In addition to the decisions in *Rosy, Econostat* and *Bosal*, discussed above, other cases where the ambit of the adaptation right was discussed, dealt with artistic work. These cases are *Cavendish Textiles Limited v Manmark Pty Ltd* 1984 115 JOC (T) and *Rapid Phase Entertainment CC v South African Broadcasting Corporation* 1996 597 JOC (W). In *The Prime Software v Commercial And Industrial Computer Systems* 1996 632 JOC (W) the court dealt with the issue of locus standi based on the ownership of copyright in modified computer programs. The case did not deal with adaptation.

jurisprudence,[391] decompilation is not treated as a form of adaptation, likely because this basis is too narrow, as the above analysis has shown. The majority of cases in foreign law focus on decompilation as a form of 'wholesale reproduction' by way of incidental or transient copying. Therefore, although the restricted act of making an adaptation is exempted in foreign law in relation to decompilation, there has been no reason advanced in case law for doing so, and no basis could be found in the legislative history of these provisions or the founding documents for copyright in computer programs.[392] This does not suggest that adaptation is irrelevant to decompilation, it merely suggests that there is no reason to pursue this enquiry any further because it has not gained acceptance, or substantial judicial consideration, in any case in SA, the US or the UK and can, therefore, make no substantial contribution.

3 2 2 2 Reproduction by intermediate copying

The final contention, that decompilation is an infringing act, is based on the fact that, in order to initiate the decompilation process, the object code is loaded into the computer and accessed by the decompiler program. The contention is that this step, the act of loading the object code, makes a copy of the entire object code.

It is true that, in order to decompile the object code with the aid of a computer, it must first be copied onto that computer and stored in digital form. This act of reproduction is non-contentious because, as long as it is an authorised copy of the program, copyright law does not prohibit the reproduction of the program code onto a computer. However, in order to decompile the object code, the decompiler program must analyse the instructions and write a new set of instructions.

Technically, this requires that the work be copied again by the computer into temporary storage in order to process the instructions. Depending on how this process is carried out, it may involve the copying of the work as a whole or in parts. Where the original work is of a substantial size, this process may be carried out in parts simultaneously by several computers or programs, sequentially in parts by the same computer or in

[391] See chapter 4 below. It will be shown that the early decompilation cases, particularly in the US, dealt with the translation issue in a similar manner to the arguments advanced above.
[392] See the analysis in paragraphs 4 2 1 and 4 3 1 below.

concert with other programs. This process may also involve further copies of the work being made in the temporary memory of the computer at different locations or on different storage media.[393]

It is possible, in the case of a sophisticated decompilation process, to restrict the process to only certain parts of the work, such as, the function statements or strings only. However, in order to identify these parts, the whole of the program must, at least superficially, be reviewed as a whole by the decompilation program. Therefore, at least one copy of the object code is made by the computer.

In all of the above cases, temporary reproductions are made by the computer as part of the decompilation process. Because the temporary storage of a computer is reused, automatically, depending on which program or task it is carrying out, the temporary copies of the object code are necessarily erased once the decompilation process is complete, so that the same, limited, temporary storage space can be allocated to facilitating new tasks. For this reason, decompilation of object code is said to involve *intermediate* copying.[394] This raises the question whether intermediate copying is, or should still be, a form of reproduction in terms of copyright law.

It is trite that, when a computer program is installed prior to use, the object code is copied onto the user's computer.[395] When the program is operated by the user, parts of the object code are also copied into the temporary storage of the computer, from where it may be accessed and executed.

---

[393] This would be the case where the decompilation process is not completed in one stage and the computer, or the decompiler program, is restarted. This was the basis of one of the arguments advanced in the cases of *Sony Computer Entertainment Inc v Connectix Corporation* (1999) 48 F. Supp. 2d 1212 and *Sony Computer Entertainment Inc v Connectix Corporation* (2000) 203 F.3d 596. Both cases are discussed in paragraph 4 2 3 10 below.

[394] See Lee (2006) *Marquette Intellectual Property Law Review* 542 for a technical explanation of this process and the fact that, during operation, parts of the code are stored in different locations.

[395] This risk, that reproduction of the object code will result in infringement, is the reason why the US Copyright Act specifically exempts this type of use. A similar provision was introduced the United Kingdom in section 50C of the Copyright, Designs and Patents Act 1988. See further DuCharme N F "Copyright Protection for Computer Software in Great Britain and the United States: A Comparative Analysis" (1987) 3 *Santa Clara High Technology Law Journal* 257 264-5.

This also involves the making of intermediate copies of the object code. But it does not amount to copyright infringement because these acts are essential to the purpose of the work, i.e. to use the program. That is made express in some copyright acts[396] and is always an authorized reproduction by virtue of the EULA[397] or the terms of an implied license.[398]

However, in the case of decompilation, the intermediate copies are not made in order to use the program for its normal purpose. Instead, the work is temporarily reproduced in order to achieve a different *purpose*.

In other words, the act of copying the code onto the computer, so that the program can be used for its intended purpose, is permitted. But the same act of copying is not permitted if the purpose of making that copy is not necessary to use the program in the normal manner. Technically, there is nothing to distinguish the act of reproduction in these cases – both involve reproduction to the same degree. The only way, therefore, to prohibit decompilation based on the act of reproduction is to limit the purpose for which the intermediate reproduction may be made.

Consequently, unless the making of intermediate copies is expressly permitted for the purpose of decompilation, either by copyright law or the terms of a licence, it will amount to the making of an infringing reproduction.

---

[396] In the UK in section 50C:

> "(1) It is not an infringement of copyright for a lawful user of a copy of a computer program to copy or adapt it, provided that the copying or adapting—
> (a) is necessary for his lawful use; and
> (b) is not prohibited under any term or condition of an agreement regulating the circumstances in which his use is lawful."

And in the US in section 117(a):

> "It is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided:
>> (1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner."

[397] The end-user license agreement.

[398] Considering that installation is a necessary pre-requisite to using a computer program for its intended purpose, it is accepted that, even in the absence of any EULA, the act of providing an authorized copy of the object code to a user would imply permission for the program to be installed.

Therefore, in countries where a decompilation exception has been granted, it deals specifically with the legal status of intermediate copies as an exempted form of reproduction. This is discussed in detail in chapter 4 below. However, in South Africa, where no decompilation exception exists, the status of intermediate copies is determined exclusively by the application of the reproduction right.

It is submitted that the mere act of copying the object code into the memory of the computer should not amount to a reproduction, regardless of the purpose for which it is made. In chapter 5 below, the argument is made that, where decompilation is concerned, it is improper for copyright law to dictate the purpose for which the decompilation may be carried out. The same basis of this argument applies here. It is improper to exempt intermediate reproduction if it is made for one form of use but not for another, unless there is a significant difference between the forms of use which will impact on the rights of the copyright owner.

But the making of an intermediate copy during decompilation poses no greater threat to the rights of the copyright owner than the making of intermediate reproductions during installation or execution of the program. As shown below, the act of decompilation does not facilitate the making of pirated copies of the original program. Piracy may be achieved without the cost and effort of decompilation. Furthermore, the act of decompilation is, as shown below in chapter 5, not primarily a means to create derivative computer programs and, therefore, does not facilitate the making of competing programs. Its primary purpose is to read the code and understand the operation of the program. This has no impact on the rights of the copyright owner. If a person decompiles a computer program and, thereafter, copies the literal text into a new program, to the extent that it amounts to a reproduction of the original source code, the copyright owner’s ability to institute a claim for copyright infringement remains unaffected. To consider the act of intermediate copying, for the purpose of decompilation, an exempted form of reproduction does not affect the scope of the copyright owner’s rights or any remedy he may have.

Thus, there is no substantive reason in copyright law why intermediate copying of the object code, for the purpose of decompilation, should be considered an infringing reproduction. However, the fact remains that decompilation involves the making of intermediate copies and, unless this form of reproduction is specifically permitted for the purpose of decompilation, the restricted act of reproduction is infringed.

As argued above, this *should not be* the legal position because, from a normative perspective, it is arbitrary. But, it cannot be submitted that decompilation *does not* involve the making of infringing reproductions because, as shown above, temporary reproductions of the object code are made during this process and such copies are, *per se*, infringing reproductions. This work returns to the question of reproduction by intermediate copying during decompilation, further below,[399] and finds an additional reason why this form of reproduction should not amount to infringement, based on the fairness analysis.

At this point, all four contentions that decompilation is an infringing act have been analysed and it has been found that, in SA copyright law, only the last contention, namely, intermediate copying, has a sound theoretical basis.

Thus, it is concluded that decompilation is prohibited in terms of SA copyright law but that the basis is significantly narrower than originally assumed. It is clear that decompilation does not amount to the making of an adaptation in any way, unless a wide meaning is attributed to copying and decompilation is considered a change in notation. Similarly, it is clear that decompilation does not amount to the making of a copy of the source code and is only a temporary reproduction of the object code.

3 3 Decompilation justified

In light of these findings, it is submitted that decompilation *should be* justified in terms of SA copyright law principles. This submission is based on a technical understanding of the decompilation process, the meaning of translation, and the way in which the element of copying should be interpreted. This is the decompilation justification, and it relies on an interpretation of copyright principles in light of the technical reality and, to some extent, normative findings about the extent to which copying should be interpreted. In order to support this view, the problems associated with a literary-analogy has been explained above to illustrate the need to give effect to the *sui generis*

[399] See paragraph 5 3 1 2 below.

classification of computer programs by interpreting copyright principles in a different manner.

However, it has been acknowledged that decompilation is currently prohibited by the SA Copyright Act. Therefore, this chapter must conclude that the act of decompilation cannot be fully justified by aligning copyright principles with the technical reality of the decompilation process, but that the act of decompilation should be justified. In order to support this argument, and arrive at a conclusion on how decompilation should be regulated by copyright law, further analysis is required.

In this respect, the *sui generis* classification of computer programs in SA law remains important because it offers the opportunity to treat computer programs differently than literary work, where it is appropriate to do so. In the rest of this work, it will be shown that it is appropriate, based on case law and international law, to consider decompilation a permissible act.

As will be shown, the meaning of copying in relation to decompilation of computer programs, has not been extended to its wide meaning. Nevertheless, the common approach in foreign law is to exempt decompilation from the meaning of adaptation and reproduction. This means there is no direct support for a restricted reading of the meaning of copying in SA copyright law, although there is also no support for a wider reading.

Consequently, the decompilation justification proposed in this chapter remains vulnerable to a wide reading of the meaning of copying for the purpose of an adaptation case, as long as SA copyright law adheres to the literary-analogy. Thus, it is necessary to also investigate how the decompilation prohibition may be addressed in SA copyright law without reliance on the decompilation justification alone.

This requires an analysis of the existing models, in foreign law, to justify decompilation as either a form of fair use or fair dealing in protected work. In the course of this analysis, the literary-analogy is evaluated further and periodic reference is made to the contents of this chapter, particularly the discussion of the technical nature of decompilation and the realignment with copyright principles suggested above. The purpose is to prove that decompilation in SA copyright law may be permitted in a manner that is both legally and technically sound, maintains a clearer divide between the idea and the expression and is better equipped to serve national interests.

Chapter 4

The Development of a Decompilation Exception in Foreign Copyright Law and the Role of Anti-circumvention Protection Measures

4 1 Introduction

In the preceding discussion it was shown that the process of decompilation is a technical series of steps that are essential in order to make the ideas, expressed in the work, accessible to the reader. It was also shown that the result of the decompilation process does not create a translation of the work and should not amount to the making of an adaptation or reproduction of the work.

The proposed conclusion, upon a technical construction of decompilation, is that the process facilitates nothing more than the exposure of the original creator's ideas. In addition, it was shown that this process does not necessarily *reveal* the ideas, because it does not reveal the source code, but merely provides an interpretation of the text that suggest what the ideas could be. In other words, decompilation creates a new set of instructions *sans copying,* which conveys the message, or the idea, in a form that a subsequent author may read and understand. By doing so, decompilation makes the ideas underlying the work accessible to another.

This analysis proved, from a technical point of view, that a decompilation prohibition is inconsistent with the principles of copyright law, in particular the idea/expression separation and should not constitute the making of an infringing reproduction. The conclusion, on the evidence above, is that the scope of copyright in computer programs must be understood to be subject to the following: **computer code is the vehicle for the protected expression and not the subject of the protection itself.**

However, this view cannot rely on a technical analysis alone. It is not sufficient to show that a prohibition on decompilation is *inconsistent* with the spirit of copyright law. It must also be shown that the process of decompilation is *consistent* with the contemporary principles of copyright law and the model for progress developed above.

In order to advance this point, the following discussion embarks on a review of selected decompilation exceptions proposed by, and implemented in, foreign law. This section seeks to illustrate the impetus for a decompilation exception and, by showing a sharp misalignment between the nature of the work, the need for access thereto and the limitations inherent to the exception, make it clear that it would be inappropriate for SA copyright law to develop in a similar manner.

In addition, the analysis of foreign law allows this work to identify the technical and legal factors that are common to a decompilation exception and which should be considered when a local exception is proposed.

4 2 The American position

4 2 1 The founding principles for copyright protection in computer programs

One of the first works, and still one of the most comprehensive and widely cited, to consider the copyright protection of computer programs, is the *Final Report of the National Commission on New Technological Uses of Copyrighted Works.*[400] The CONTU final report was commissioned "to assist the President and Congress in developing a national policy for both protecting the rights of copyright owners and ensuring public access to copyrighted works when they are used in computer and machine duplication systems, bearing in mind the public and consumer interest."[401]

The voluminous report, dated 1979, is still frequently cited with authority in American jurisprudence and is, surprisingly, the earliest authority to deal with decompilation in detail. The need to address this issue is anticipated by CONTU from the outset. The report states that the foundation of its recommendations rely on a number of core principles.

[400] CONTU (1978) *Final Report of the National Commission on New Technological Uses of Copyrighted Works*. The full text of the CONTU Report is available at http://digital-law-online.info/CONTU/contu-toc.html (accessed November 2019).

[401] CONTU *Final Report* 1.

It states:

> "To provide reasonable protection for proprietors without unduly burdening users of programs and the general public, the following statements concerning program copyright ought to be true:
>
> 1. Copyright should proscribe the unauthorized copying of these works.
> 2. Copyright should **in no way inhibit the rightful use** of these works.
> 3. Copyright should **not block the development** and dissemination of these works.
> 4. Copyright should **not grant anyone more economic power** than is necessary to achieve the incentive to create."[402]

These, seemingly obvious, statements neatly express the purpose of an exception to copyright protection and provide a comprehensive test against which the postulated conduct may be measured in order to determine whether or not it amounts to justifiable use of the work. In other words, it is an expression of the values that underlie a fairness analysis. It is applicable both in the case of an infringement defence and during a consideration of the scope of protection afforded to a work.

Consequently, the four values expressed by CONTU make it clear that the protection of computer programs shall be governed by copyright law only to the extent that it does not inhibit the use of the work to create further works, whether or not the subsequent work amounts to a derivative or adaptation of the original.

This crucial limitation on the protection of computer programs is explained by CONTU with reference to the fact that, in order to use a program for its primary purpose, the program must be installed on the user's computer.[403] Because this necessarily

[402] CONTU *Final Report* 12 (emphasis added).

[403] It is noted that this statement is no longer universally true. Remote access and cloud-based computing has made it possible to use a computer program without the need to install the source code on the user's computer. At the time of the CONTU Final Report, cloud-based computing was unknown. This technical advance has no impact on the legal argument made above. Furthermore, although it is true that, in the case of web-based applications, the whole of the executable code is not copied onto the user's computer, it may still be argued successfully that, at least, a temporary reproduction of a substantial part of the work is made by the user's computer during the operation of the application. This reproduction may be made by the browser onto the temporary storage of the user's computer or by the

involves the making of a reproduction, the report submits that a user "shall be provided with a legal right to copy it to that extent which will permit its use by that possessor."[404]

The final report then proceeds to define the scope of "use" in the above context. It finds that, in addition to the necessary initial installation of the program, other justified uses of the program include the process of converting the program into another language and, notably, make changes to the code of the work in order to use it.[405]

These justified uses are circumscribed by CONTU in light of prevailing realities of software at the time,[406] namely, the absence of a standard programming language or means to express ideas for software and the likelihood that a user may wish to adapt the program in order to allow it to function on their computer.

This makes it clear that, well before compilation of computer programs, and the subsequent dissemination of work in object code form, became common practice, it was considered improper for copyright law to extend to the reproduction of the code for any purpose other than piracy.

The final report is unambiguous on this point and goes as far as establishing a "right of adaptation"[407] for this purpose. It states:

> "The copyright law, which grants to copyright proprietors the exclusive right to prepare translations, transformations, and adaptations of their work, should no more prevent such use than it should prevent rightful possessors from loading programs into their computers."[408]

---

remote server, at the behest of the user, in order to permit the use or, most commonly, by both the user's computer and the server.

404 CONTU *Final Report* 13.

405 CONTU *Final Report* 14. "The conversion of a program from one higher-level language to another to facilitate use would fall within this right, as would the right to add features to the program that were not present at the time of rightful acquisition."

406 CONTU *Final Report* 13.

407 14.

408 14.

Consequently, the act of altering the code does not fall within the scope of the restricted act of making an adaptation. Even if decompilation is incorrectly considered a form of adaption *in fact*[409] the act of amending the code during the process of decompilation does not amount to an adaptation *in law*. The fact that the CONTU final report only deals with amendment to, and conversion of, source code, has no impact on this finding. At the time of the report, the majority of programs were distributed in source code form. Copyright law now treats source and object code as analogous.[410] Thus, the suggestion of the CONTU final report in relation to source code holds true for programs in object code. It would be improper to suggest that the final report permits the adaptation of source code by the user, but does not support this form of justifiable use of the object code.

Furthermore, the commission recognised the likelihood that future development will necessitate the expansion of the right of adaptation and charged the judiciary with the responsibility to expand on the scope of "use" outlined by CONTU. It went as far as anticipating a situation which "permits future infringers to use an author's program without copying"[411], i.e. decompilation.

However, the CONTU report is not authority for a blanket permission to decompile. As noted above, it established the user's right to adapt the source code in order to make the program perform its intended function (either by changing the programming language or by amending the code). It did not, and could not, anticipate the need to carry out a technical process, such as decompilation, in order to read or understand the work. At the time, the barrier imposed by compilation did not exist. Nevertheless, the report made it clear that in certain circumstances the code may need to be adapted by the user and that such conduct was not a matter for copyright law or, if it was, would be justified as an instance of prima-facie authorised use.[412]

---

[409] See the discussion in paragraph 3 2 1 *et seq.* in chapter 2 above.

[410] TRIPS Article 10(1).

[411] CONTU *Final Report* 23.

[412] CONTU *Final Report* 14: "[A] right to make those changes necessary to enable the use for which it was both sold and purchased should be provided."

In short, the CONTU report makes clear, in no uncertain terms, that the functional nature of computer programs will require that copyright principles be purposefully interpreted so that this type of work will not be subject to wider protection than any other type. In particular, the report considers it essential that computer programs shall be available for analysis by the licensee.[413]

4 2 2 The statutory position

Despite the sound approach in the CONTU report, the development of copyright law through case law is a picture of ever-increasing expansion on the scope of protection. This is partly due to the inherent limitations of the report which, because of the early stage at which it was drafted, left it open for parties and courts to ignore the general sentiments outlined above. As a result, when faced with seemingly unique or highly technical aspects of computer programs, the courts were persuaded to expand the scope of protection or maintain the widest possible interpretation.

The second reason is the manner in which the recommendations of the CONTU report were incorporated into federal law. The report led to the Computer Software Copyright Act of 1980,[414] which amended title 17 of the United States Code,[415] by introducing a definition of computer programs[416] in §101 and replacing §117 in its entirety to make provision for certain narrow exceptions.[417] However, it did not include computer programs in the list of eligible types of work in §102 or in the definition of literary works in §101.[418] This means that, although computer programs are mentioned in US

---

[413] See the footnote above with reference to CONTU *Final Report* 14, which suggests that where a computer program is bought with the intention to analyse it, the licensee would have a right to make the necessary changes to the program to carry out the analysis.

[414] Computer Software Copyright Act of 1980 Pub. L. No. 96-517, 94 Stat. 3015, 3028.

[415] Copyright Act of 1976, Pub. L. No. 94-553, 90 Stat, 2541, Title 17 of the United States Code (US Copyright Act).

[416] The wording of which reads:

> "A "computer program" is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result."

[417] §117 makes provision for the authorized adaptation of computer programs during installation or maintenance, the reproduction thereof for archival purposes and the lease or trade in such authorized copies or adaptations provided that it divests the original licensee of all rights.

[418] A literary work is defined as:

Copyright Act, it is left drifting.[419] It is a uniquely defined type of work subject to some limitations, but without any specific provision about the nature or scope of its protection.

In order to cure this defect, the early courts needed to perform a two-stage reading-in exercise. In the first place, it had to expressly categorise computer programs as a type of literary work and then determine its scope with reference to existing copyright law before the 1980 amendment based on so-called "legislative history."[420] In particular, it relied on the comments by the House Judiciary Committee,[421] made in reference to the 1976 amendment, which did not contain any of the above amendments regarding computer programs. As a result, the court held that "the copyrightability of computer programs is firmly established after the 1980 amendment to the Copyright Act."[422]

> "Works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied."

[419] See *Apple Computer v Formula International Inc.* (1983) 562 F. Supp. 775 at 11 *et seq.* where the court finds that the 1980 amendment did, nevertheless, provide sufficient certainty on the nature of protection, albeit not on the scope. The Register of Copyrights referred to the introduction of this act at the time, but provided no guidance on its application. See Librarian of Congress (1980) *83rd Annual Report of the Register of Copyrights* at 12.

[420] *Williams Electronics Inc v Artic International Inc* 685 F.2d 870, 215 USPQ 405 (3rd Cir. 1982) at 24 and fn4; *Apple Computer Inc v Franklin Computer Corporation* 714 F.2d 1240, 70 A.L.R.Fed. 153, 219 USPQ 113 (3rd Cir. 1983) at 30-34. See also Patry W F *Copyright Law and Practice* 2ed (1994) 89-120 and 2-13 of the 2000 Cumulative Supplement to Chapter 1 for a summation of all of the amendments to US Copyright Law before, during and after the 1980 amendment.

[421] House Judiciary Committee *Copyright Law Revision House Report* No. 94-1476, 94th Congress, Second Session at 54:

> "The term "literary works" does not connote any criterion of literary merit or qualitative value: it includes catalogs, directories, and similar factual, reference, or instructional works and compilations of data. It also includes computer data bases, **and computer programs** to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves" (emphasis added).

[422] *Williams Electronics v Artic International* at 24.

This finding was subsequently confirmed,[423] even though the courts did not consider whether their reading of the amendment (which is based on pre-amendment law) is a full and accurate reflection of the founding document, namely the CONTU report.

This created the erroneous assumption that because the 1980 amendment may be said to have classified computer programs as literary works, it also determined the scope of copyright in computer programs. While both courts correctly identified the nature of the work (as a type of literary work), they erred in their assumption that the 1980 amendment intended for this type of work to be subject to all the restrictions imposed on the exploitation of literary works.

As shown above, the 1980 amendment did not describe the scope of protection and the CONTU report made it clear that, "should a line need to be drawn to exclude certain manifestations of programs from copyright, that line should be drawn on a case-by-case basis by the institution designed to make fine distinctions – the federal judiciary."[424] Furthermore, the reliance on the House Judiciary Committee Report established in case law the assumption that copyright in a computer program *prima facie* extends to all of the code. This is because both reports cited by CONTU state that the requirements for subsistence of protection in "section 102(b) is intended, among other things, to make clear that the expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of the copyright law."[425]

This led to the mistaken belief that copyright in the program *extends to all of the code*, being the protected expression as understood for literary works, rather than those parts of the code that are original. Consequently, the manner in which protection for computer programs was introduced and, from the outset, misapplied, established the principle that, save for the underlying processes or ideas, everything that is written in code is protected in exactly the same way that words in a book are protected. As a

[423] *Apple Computer Inc v Franklin Computer Corporation* at 34.

[424] CONTU *Final Report* 22-3.

[425] CONTU *Final Report* 2.

result, all subsequent case law on computer programs had to grapple with the need to adjust the scope of protection by way of exceptions. As the CONTU report states:

> "[T]he many ways in which programs are now used and the new applications which advancing technology will supply may make drawing the line of demarcation [between the copyrightable form of a program and the uncopyrightable process which it implements] more and more difficult. To attempt to establish such a line in this report written in 1978 would be futile."[426]

And yet, the American position incorporated such a demarcation into its law from the outset.

The third reason why case law developed an overly-protectionist approach to computer programs is the literary-works analogy – a consequence of the misapplication outlined above. In a vein remarkably similar to the South African authorities, contemporary commentators in the US argued, correctly, that computer programs are a type of work that is suitable for copyright protection.[427] However, in a similarly problematic approach to South Africa, it found authority for this view in the perceived literary nature of computer programs. In other words, not only are computer programs to be protected in a manner similar to literary works, it shall be treated exactly the same as literary works. This made it difficult, and at times impossible, to justify any inherent limitation on the scope of computer programs if such a limitation is not also already evident in the scope of protection afforded to other instances of literary work. It also meant that any argument about the functional nature of computer programs, and the impact this should have on delineating copyright law, could not be entertained. In short, if it does not work for traditional literary works, it cannot work for computer programs.

The reverse argument is also true – any leeway granted to the user of a computer program is only justifiable if the user of another type of literary work is entitled to the same in relation to that work. If not, the use must be prohibited. Thus, for example,

[426] CONTU *Final Report* 22.

[427] Miller *Harvard Law Review* at 982 provides a concise overview of the prevailing arguments at the time.

one cannot allow the decompilation of a program because it looks like the making of a translation, which is a restricted act in relation to literary works.

This tie between computer programs and literary works is described in this work, for the sake of convenience, as the literary-works analogy-approach, or the literary analogy, explained in chapter 2 above. Despite the fact that it created problems for the court from the outset, some scholars made extreme efforts to align every aspect of computer programs with an analogue equivalent:

> "Just as no two novelists independently would compose the same detailed plot of the downfall of a tragic hero, or the travails of star-crossed lovers, or the consequences of making a pact with the devil, no two programmers independently would design a program that enabled the computer to solve highly intricate problems with the same structural details, let alone precisely the same set of instructions. Furthermore, the communicative precision required of a computer programmer is not unlike the discipline that a poet must achieve to convey a complex message within the confines of a tightly constrained meter or that of a composer who must work within the limited ranges of musical instruments or of the human voice. In each case, the copyright law rewards the author's imagination and originality of expression in the hope of encouraging further creative productivity."[428]

In these words, one finds all of the problems created by the analogy-approach. It explains the problematic application of the merger doctrine to computer code,[429] the confusion of the idea/expression separation when it comes to program structure and layout, the move away from protection of original expression toward protection of all coded instructions, an abandonment of the functional nature of computer programs and the application of copyright law to non-literal elements.

These sentiments lay bare the core of all of the mistakes made by the courts that led, inevitably, to the prohibition on decompilation and the need for a decompilation

[428] Miller *Harvard Law Review* 984 (emphasis added).

[429] See Scott M D *Scott on Information Technology Law* 3ed (2019)2-101; See also Samuelson P "Functionality and Expression in Computer Programs: Refining The Tests For Software Copyright Infringement" 2016 *Berkeley Technology Law Journal* 31 (2) 1215 1278-1281.

exception. The undesirable impact this would have on the construction of copyright law in South Africa, has been detailed in the preceding chapter.

In what follows, selected foreign case law is reviewed to illustrate these points and make clear where the decompilation exception proposed in South African law comes from and why it is ill-suited to the South African context.

4 2 3 Case law

The following discussion deals with a number of cases. Some effort has been invested in the selection of cases to be comprehensive on the issues under discussion. However, it must be noted that the cases under review were selected based on their relevance to the delineation of the scope of copyright in computer programs and, specifically, those cases that influenced the current position on decompilation. Thus, the case law below is not merely a listed discussion of all decisions that deal with computer programs in general or reverse engineering of other types of work. Furthermore, the discussion includes the earliest cases on the scope of protection, but purposefully omit those decisions that pre-date the 1980 amendment (with one exception) and which dealt with the protection of computer programs or video games in an indirect manner by way of audio-visual works,[430] or those cases that dealt only with the reproduction of software by way of physical reproduction of the storage medium or ROM chips.[431] For the sake of completeness, a list of the cases that were canvassed but omitted from this work on these bases, is provided in the preceding footnotes, arranged by topic.

---

[430] *Midway Manufacturing Co v Dirkschneider* 543 F. Supp. 466 (D. Neb. 1981); *Atari Inc v Amusement World Inc* No. Y-81-803 (D. Md. Nov. 27, 1981); *Midway Manufacturing Co v Omni Video Games Inc* 668 F.2d 70 (1st Cir. 1981); *Midway Manufacturing Co v Artic Internationa, Inc.* No. 80 C 5863 (N.D. Ill. March 10, 1982); *Stern Electronics Inc v Kaufman* 669 F.2d 852 (2d Cir. 1982); *Atari Inc v North American Philips Consumer Electronics Corp* 672 F.2d 607 (7th Cir. 1982). All of these cases are cited, and dismissed where not applicable, by the court in *Williams Electronics Inc v Artic International Inc*, which is discussed further below.

[431] *Data Cash Systems Inc v JS&A Group Inc* 480 F. Supp. 1063 (N.D. Ill. 1979); *Tandy Corp v Personal Micro Computers Inc* 524 F. Supp. 171 (N.D. Cal. 1981); *Data Cash Systems Inc v JS&A Group Inc* 628 F.2d 1038 (7th Cir. 1980).

4 2 3 1 *Synercom Technology v University Computing Company (1979)*[432]

This was the first case to consider copyright in relation to computer programs after the publication of the CONTU report and dealt with the law as it stood before the 1980 amendments.[433] However, it is included here because some of the findings of the court had a direct impact on the interpretation of the law after 1980.[434]

The dispute related primarily to copyright in the printed instruction manuals for two computer programs, namely, STRAN, created by the plaintiff Synercom, and SACS II, created by the defendant Engineering Dynamics Incorporated (EDI) and distributed by University Computing Company (UDI). Both programs are structural-analysis programs designed to carry out complex calculations in the engineering sector.

The STRAN program was based on the IBM FRAN program and used its underlying algorithm, but made extensive amendments and additions in order to, inter alia, make STRAN compatible with non-IBM computers, increase its computing capacity and perform its tasks more efficiently.[435] Significantly for this dispute, Synercom developed a unique[436] and user-friendly way to enter data into the STRAN program. This took the form of a data format – a predetermined and specific selection and arrangement of variables dictated by Synercom. Synercom created nine classes of input formats, all of which received copyright registration in 1976.

The formats are described by the court as follows:

> "[I]nput formats express to the user the sequencing of data for simplified access to the computer programs. The formats by their placement of lines, shaded art,

[432] *Synercom Technology Inc. v University Computing Company and Engineering Dynamics Inc.* 1978 462 F. Supp. 1003 (*Synercom*).

[433] The case was decided in terms of the Copyright Act of 1909 (An Act to Amend and Consolidate the Acts Representing Copyright (1909) Pub.L. 60–349 35 Stat. 1075) because the facts, and registration of the works, occurred before the Copyright Act of 1976 (An Act for the general revision of the Copyright Law, title 17 of the United States Code, and for other purposes Pub.L. 94–553 90 Stat. 2541) came into effect on 1 January 1978.

[434] See the discussion below of the *Williams* judgments and the *Whelan* case in particular.

[435] *Synercom* at 9.

[436] At 9.

and words tell the user what data to place where and how to do it. It **communicates the selection arrangements and the sequence**."[437]

These improvements and, in particular, the creative and more accessible input formats (in later case law referred to as interface specifications), established STRAN in the market. The particulars, about how the input formats worked and how it should be used, were published by Synercom in a series of manuals and subject to copyright protection.

In order to compete successfully[438] in the market, EDI ensured that SACS II would be compatible with data inputs arranged according to the format set by Synercom. In this way, it achieved interoperability between SACS II and STRAN in an indirect and one-directional manner.[439] Particulars about how to use SACS II were published in a series of manuals which contained "mirror images of some of the input cards and instructions [which] effectively enabled a customer to use the STRAN input format."[440] The SACS II manuals also contained other text (not related to the formats), large portions of which were verbatim reproductions of the STRAN manuals.

Synercom contended that copyright was infringed in: (1) the text of the manuals not related to the formats; and, (2) the formats as published in the manuals, upon which the SACS II program was built. The court found in favour of Synercom on the first issue and against on the second issue. The first issue related to reproduction of standard text and was non-contentious. It is not relevant to this work.

On the second issue, it declared that the reproduction of the formats did not amount to infringement because "order and sequence are **expressed ideas, not expressions**"[441] and EDI was, thus, free to use the ideas for their own program and in their manuals.

437 At 23.

438 *Synercom* at 12-13.

439 At 13.

440 At 14.

441 At 31 (emphasis added).

This finding is explained by the court as follows:

> "If the idea is the sequence and ordering of data, there was no infringement. If sequencing and ordering of data was, however, expression, it follows that EDI's preprocessor program infringed. As earlier suggested and as will be demonstrated, Synercom's argument is double-edged. **If sequencing and ordering is expression, what separable idea is expressed?**"[442]

As a result, the court rejects "Synercom's argument that the order and sequence of data was the expression, not the idea,"[443] which leads it to make an alternative order to the effect that, if it is wrong in finding that EDI merely reproduced ideas, EDI was nevertheless free to use the formats because "formats are not copyrightable."[444] In other words, the court finds that *in casu* the formats were not protected expressions and, in the alternative, all formats are non-original expressions.

The court does not explain the difference between "expressed ideas" and "expressions"[445] but it seems to suggest that formats are unprotectable (i.e. so-called expressed ideas) because these are statements of fact. The court supports this finding by firmly establishing the literary analogy in copyright case law for computer programs. It argues that Synercom may not claim copyright in the formats because, if it was allowed to do so, "it would follow that translating the expression of the [Synercom] manual to FORTRAN to another manual equally would be an infringing use."[446] In other words, if a format was incorporated in the code of a computer program (as happened in this case with the creation of SACS II) and a manual for that program is, thereafter, published, the manual would be an infringing *translation* of an infringing translation.

Thus, according to the court, the SACS II program would be an infringing work in relation to Synercom's format and, indirectly, the STRAN program. The court finds

---

[442] *Synercom* at 27 (emphasis added).

[443] At 28.

[444] At 32.

[445] At 31.

[446] At 31 fn5. FORTRAN is the programming language used by both Synercom and EDI.

such a prospect untenable because it falls outside the meaning of translation as it applies, in the opinion of the court, to computer programs:

> "[It] is as clear an infringement to translate a computer program from, for example, FORTRAN to ALGOL, as it is to translate a novel or play from English to French. In each case the substance of the expression (if one may speak in such contradictory language) is the same between original and copy, with only the external manifestation of the expression changing. Likewise, it would probably be a violation to take a detailed description of a particular problem solution, such as a flowchart or step-by-step set of prose instructions, written in human language, and program such a description in computer language. **But here the similarity to literary translation ends**."[447]

Therefore, the court finds that EDI did nothing more than prepare a computer program based on general descriptions (namely the formats) contained in the STRAN manuals. And, since the court had already rendered the formats mere ideas, it cannot be said that EDI translated a protected work when it created the SACS II program, or its manuals.[448] To find otherwise would be to trigger the landslide of prohibited translation expounded by the court.

This decision has been roundly criticised[449] and, in many respects, overturned by subsequent judgments.[450] However, some problematic aspects remain in

---

[447] *Synercom* at 31 fn5.

[448] Curiously, in support of this finding the court then treats the formats as *protected* works, but find that EDI nevertheless did not infringe because it created an independent work based on the idea embodied by the format (as opposed to reproducing the expression). In the following extract, the court refers to "statement" and "description" to mean the formats. See *Synercom* at 31 fn5:

> "The program and the statement are so different, both in physical characteristics and in intended purpose, that they are really two different expressions of the same idea, rather than two different versions of the same expression. Hence EDI's preparation of a FORTRAN preprocessor program from the descriptions contained in the manuals cannot constitute an infringing derivative use provided this was done without copying of the plaintiff's FORTRAN program, as it was."

[449] See for example: Barfield W and Blitz M J *Research Handbook on the Law of Virtual and Augmented Reality* (2018) 176; Galler B A *Software and Intellectual Property Protection: Copyright and Patent Issues for Computer and Legal Professionals* (1995) 19-20; Scott M D *Scott on Information Technology Law* 3ed (2019) 2-94 to 2-96.

[450] Most notably by the court in *Whelan Associates Inc v Jaslow Dental Laboratory Inc (1986)* at 1239-1240; See also *Engineering Dynamics Inc v Structural Software Inc* (1994) 26 F.3d 1335 at 19 (modified

jurisprudence and are relevant to the development of the decompilation exception in American law.

First, the judgment entrenched the view that computer programs are to be treated as nothing more than literary text. In particular, its opinion on the translation of a computer program from one high-level language to another is directly responsible for the later view that encoding instruction in high-level language (source code) to low-level language (object code) is also a translation within the literary meaning of the word.

Second, its highly problematic commentary on the likelihood that the act of creating a program, based on the detailed description of a program drafted by another, would amount to the making of an infringing copy (by translation), signals a fundamental misconception about what computer programs are, which persists to this day. It suggests that the content and structure of all computer code is eminently predictable and singular because code shares, in the view of the court, all the strictures of human language that make it capable of direct translation.

Consequently, it would be an infringement, through translation, to write code based on another's description because the result must necessarily conform with predetermined rules. While this is true to some extent regarding structure, it is not true for the content.[451] The sentiment of the court, however, conflates the strictures of programming language (which determine some elements of the structure) with the idea that the content is merely a consequence thereof. This elevates in importance

---

by *Engineering Dynamics Inc. v Structural Software Inc* (1995) 46 F.3d 408). See also *Williams v Arndt* (1985) 626 F. Supp. 571 which, while not referring to the *Synercom* case, expanded the meaning of translation of computer code suggested by the court. See contra *Plains Cotton Cooperative Association v Goodpasture Computer Service Inc* (1987) 807 F.2d 1256 at 1262 which was said to support *Synercom*. However, according to the same court in *Kepner-Tregoe Inc v Leadership Software Inc* (1994) 12 F.3d 527 fn 20, it did not, and found that *Synercom* "is binding neither in its legal holding nor by compelling factual analogy."

[451] See also the discussion in Scott *Scott on Information Technology Law* at 2-97 where the author explains the impact of this finding in *Synercom* in relation to detailed flow-charts and the problem it creates. Scott argues, correctly, that it differentiated between the rights of a user in relation to a description of a computer program and the description of another product by, for example, using a drawing of the product. This distinction, it is submitted, had no merit.

the structure of the program (the sequence and arrangement of lines of code) above the content (the text and function of each line of code).

It is true that, at the time of this judgment, existing programming languages did impose far greater limitations on the structure and content when compared to the flexibilities offered by modern programming languages.

While this, the standardisation effect of programming language, might excuse the court's mistake in this case, it is no longer an acceptable reason to prevent the so-called translation (i.e. decompilation) of code just because the structure itself is, in some cases, a protectable expression.[452]

Unfortunately, this mistaken idea remains present in subsequent case law to the extent that infringement actions continue to focus on the factual reproduction of code. It placed an extreme and undue emphasis on the literal manifestation of the work, which has its origin in this case. In the case of reverse engineering matters, the impact is particularly evident. It would take many years, and difficult development, to counteract the focus on factual reproduction with a degree of consideration paid to the originality and purpose of the work.

The third problem with this case is related to the predictability error discussed above, but relates to the court's finding about the content of the formats. The court finds that the unique selection and sequence of inputs which Synercom expressed in the formats do not amount to a copyrightable work because it is merely an idea. This idea, the court suggests, may be original to the plaintiff, but it may not be monopolised because "[if] sequencing and ordering is expression, what separable idea is expressed?"[453]

The reasoning of the court is that the data formats represent an idea which lacks "stylistic creativity above and beyond the bare expression of sequence and

[452] *Synercom* is correctly credited as the origin of a standardization defence to copyright infringement. See Edgar S L *Morality and Machines: Perspectives on Computer Ethics* (2002)143.
[453] *Synercom* at 27.

arrangement,"[454] which would otherwise constitute a protectable expression. Therefore, the formats are nothing more than the idea or principle for a method to enter data in a computer.[455] While this might be true in the "blank form"[456] situation, it was not true on the facts. The court itself calculated that there could be over 3.5 million different expressions[457] of the selection, sequence and arrangement of variables contained in the formats and yet, EDI chose to use the exact same variations published by Synercom.

This alone should have made it clear to the court that the formats are protected expressions. Unfortunately, it was convinced, even while discussing the variability of data formats, that EDI's actions should be excused. It proceeds to do so based on a misapplication of the idea/expression dichotomy discussed above.

However, this is not the real reason for the court's finding. Important for this work is the underlying impetus for the court's extraordinary efforts to disqualify data formats from protection. In an attempt to illustrate the idea/expression dichotomy, the court draws an analogy between a data format and the figure-H pattern of a manual transmission lever in a car. It argues that the H pattern is, like Synercom's formats, susceptible to copyright protection when expressed in writing or fixed in some other manner. However, "the copyright protects copying of the particular expressions of the pattern, and does not prohibit another manufacturer from marketing a car using the same pattern. Use of the same pattern might be **socially desirable**."[458]

---

[454] At 30.

[455] *Synercom* at 31. See also at 30 where the court states that "in the usual case sequence, choice, and arrangement have only stylistic significance, rather than constituting as they would here, the essence of the expression." This seems to suggest that the court considers the forms to be stylistically significant (an idea) but not stylistically creative (a protectable expression).

[456] *Synercom* at 31.

[457] *Synercom* at 25 the court states:

> "By varying only the order constituent of the format instruction, the manner of communicating with the computer may be expressed in ten factorial (10-9-8-7-6-5-4-3-2-1); that is 3,628,800 expressions."

[458] *Synercom* at 28.

This makes it clear that the idea/expression dichotomy was applied in this case as a scapegoat while the court in fact intended to make a finding that "gave priority to compatibility and competition."[459] This established the pro-competitive rhetoric in copyright law in relation to computer programs[460] that would eventually lead to the interoperability limitation to decompilation.

4 2 3 2 *Williams Electronics v Artic International (1981)*[461]

In this case, the plaintiff (Williams) claimed copyright infringement in the computer program DEFENDER, which was stored on the circuit boards of a coin-operated video game. At the time, Williams held three copyright registrations in relation to DEFENDER. Two related to the audio-visual works and the third to the computer program which produced the effects.[462] The defendant (Artic) bought a number of circuit boards from a third party and distributed these to the public in the form of kits.[463] When the Artic boards were operated, it caused the computer program that was stored in the memory device or ROM attached to the circuit board to produce a result.

The Artic boards produced a result entitled DEFENSE COMMAND, which the court found to be "substantially identical"[464] to the result produced by the Williams program.

[459] Barfield and Blitz *Research Handbook on the Law of Virtual and Augmented Reality* 176.

[460] The sentiment is expressed by the court in *Apple Computer Inc v Franklin Computer Corporation (1982)* 545 F Supp 812 at 42 where it states that an injunction against the infringer who sought to compete would have "a devastating effect". See also the argument in support of the *Synercom* judgment expounded by Reznick A E "Synercom Technology, Inc. v. University Computing Co.: Copyright Protection for Computer Formats and the Idea/Expression Dichotomy" 1980 *Rutgers Computer & Technology Law Journal* 8 65 73-4 and the discussion of the expansion of the *Synercom* judgment to non-literal reproduction in Samuels E "The Idea-Expression Dichotomy in Copyright Law" 1989 *Tennessee Law Review* 56 321361 and 364. See also the observations in Menell P S "An epitaph for traditional copyright protection of network features of computer software" 1998 *The Antitrust Bulletin* 43 651 at 651-2 about the subsequent trend in 1980s case law to consider the competitive and commercial objectives of copyright in relation to software.

[461] *Williams Electronics Inc v Artic International Inc* 1981 U.S. Dist. LEXIS 17856 (*Williams I*).

[462] *Williams I* at 11.

[463] *Williams I* at 12 to 13. Greater detail about the actions of Artic is found in the appeal judgment *Williams Electronics Inc v Artic International Inc 685 F.2d 870, 215 USPQ 405 (3rd Cir. 1982)* at 6, discussed further below.

[464] *Williams I* at 13.

It arrived at this conclusion based on the facts that: (1) the Artic program contained an error that was present in an early version of the Williams programs;[465] (2) the Artic program contained results produced by the Williams program in the form of recorded high scores achieved during the development of the Williams program;[466] (3) the Artic program produced the identical diagnostic visual aids produced by the Williams programs;[467] (4) the Artic program contained at least 85% of the same lines of code (which the court calls "listings" in machine language);[468] and, (5) the Artic program contained the exact wording of a copyright notice embedded in the program by Williams and which still identified Williams by name as the party exerting rights in the work.[469]

The judgment does not discuss the nature or scope of protection for computer programs and is, therefore, of limited value. However, it contains one very important observation on the facts of the matter, which impacted on subsequent case law.

Based on the plaintiff's evidence, the original work was "written in **assembly language** and **includes** approximately 10,000 lines of **machine language** computer instructions."[470] It is important to note the difference between assembly language, which is a low-level programming language that must be assembled (a process similar to compilation) before it can be executed by a computer, and machine language, which represent instructions that are carried out directly by the computer. The fact that Williams' program "included" approximately 10 000 lines of machine language, means that the source code (in assembly language) for the program relied on existing work (which was previously assembled into machine language) and added to it. This was common for arcade-style video games at the time and a necessary consequence of the fact that the program was integrated in the circuit board. The existing instructions related to default functions of this type of program that would, for example, retrieve certain audio-visual works from the memory device, direct the operation of the

465 At 13.

466 *Williams I* at 13-14.

467 At 14.

468 At 14-15.

469 At 15.

470 At 11 (emphasis added).

microprocessor or the peripherals such as the coin-reader, CRT screen and loudspeakers.

The additional instructions created by Williams relied on these known, standardised instructions for this type of work and created additional instructions that relate to the unique way in which the DEFENDER program will operate. Thus, the code for the DEFENDER game embodied both original instructions and prior art. There is no evidence to suggest that Williams was the author, or the copyright owner, of the existing instructions.

The amount of work that Williams may lay claim to as an original expression is not clear. The only evidence in this respect is the submission that "[in] the memory devices the computer program requires approximately 28,000 listings"[471] or lines of code *in machine language*. Included in this total is the approximately 10 000 lines of machine language code, and however many lines were added by Williams specifically for the DEFENDER program. Since each line of code in assembly language usually result in a single line of code in machine language, one may deduce that the pre-existing code relates to 10 000 lines and Williams' original code to 18 000 lines.

This places the court's finding of "identical similarity" at issue. The computer programs in question were stored in several chips (or memory devices). When Williams carried out a process of decompilation on the work of Artic, it found an 85% similarity across all of the memory devices, when considered as a whole.[472] Only in relation to a single chip did it find absolute similarity.[473] This means that the court made its finding of substantial reproduction based on 85% of the total code and without separating the 18 000 lines, to which Williams could lay claim, from the 10 000 lines in which it did not, and likely could not, prove ownership.

Even if the court did isolate the original work and compared only that to the infringing work, it is unlikely that it would have arrived at a different conclusion. The discussion

[471] *Williams I* at 10.

[472] At 14.

[473] At 14-15.

of the two works in this case, and the subsequent appeal, make it clear that a substantial reproduction of the original elements in the work did occur.

However, the opposite might just as well have been true if Artic created a different game. There is no reason to believe that the court would have found Artic innocent of infringement under such circumstances. On a hypothesis that the 18 000 lines of original code differed entirely between Williams' and Artic's works, the method applied by the court in this case would have found that at least 36% of the works are still identical. Since the court made no attempt to separate protected lines from unprotected lines, it would likely have considered it substantial enough that more than a third of the work was copied.

This case is the first in a long line of judgments that allowed a copyright claimant to exercise his rights over the totality of the textual expression regardless of its nature. It illustrates a trend of obscuring the original contribution within a larger work consisting of unprotectable elements. As such, this judgment proves that the literary-analogy had, from the outset of statutory protection, a detrimental effect on the scope of copyright in software.

4 2 3 3 *Williams Electronics v Artic International (1982)*[474]

This case is an appeal on the judgment discussed above, brought by Artic as the plaintiff-appellant against the injunction against further copyright infringement of Williams' computer program. The facts are the same as above. However, in this case, Artic did not dispute the validity of Williams' copyright but, instead, asked the court to consider the validity and scope of protection.[475] Therefore, it is the first case that dealt with the interpretation of the Act to a substantial degree. In order to satisfy the burden

[474] *Williams Electronics Inc v Artic International Inc* 685 F.2d 870, 215 USPQ 405 (3rd Cir. 1982) (*Williams II*).

[475] *Williams II* at 7 and 8-9.

of proof that Williams' copyright registrations imposed on Artic,[476] it instituted appeal on eight grounds. Of these, only two are relevant to this analysis.[477]

The first is the contention that "the computer program would be infringed only if an unauthorized copy of the program text was made,"[478] and, because it bought the boards containing the program from a third party rather than make the reproductions, Artic could not be held liable for infringement. The court dismissed this argument as a question of fact which, by agreement between the parties, had already been settled. The court points out that "we must sustain the district court's order unless it can be challenged on some basis other than who is responsible for the copying or **how the copying was effected**."[479]

At this stage, the 1980 amendments had been enacted, which contained the exemptions relating to installation, backup and maintenance of software suggested in §117. Therefore, the court did not have these forms of use in mind. This suggests that the court indicated, in the highlighted passage, that there may be no other circumstances under which the manner in which the copy was made, would absolve the defendant.

It is doubtful whether the court in 1982 postulated a limitation on the scope of copyright protection that would leave room for reverse engineering by automated means, i.e. decompilation, and sought to rule that out. However, it certainly confirmed that the act of *textual reproduction* is a matter of fact, and not law, which must be satisfied in order for infringement to occur. The effect of this on the development of copyright jurisprudence is clear – it established the view that copying of computer code is *per se* a reproduction and that it does not matter how the apparently similar work was created, even if it was automated or by educated guesswork.

---

[476] In terms of §410(c) of the US Copyright Act the registration of copyright establishes prima facie proof of the *validity* of copyright in the work and, consequently, placed the burden on Artic to establish that the protection is unfounded.

[477] The remainder, all of which were also dismissed, related to the copyright in audio-visual works, Artic's wilful or innocent intention and the protection or reproduction of ROM chips.

[478] *Williams II* at 17.

[479] At 18 (emphasis added).

The second contention by Artic was that the scope of protection extends only to source code and not to object code. It submitted that a reproduction of the object code of a computer program does not amount to the making of an infringing copy because a copy "must be intelligible to human beings and must be intended as a medium of communication to human beings".[480] The court dismissed this argument with reference to the definition of "copying" in §101 which includes "any method now known or later developed, *and from which the work can be perceived*, reproduced, or otherwise communicated, either directly or *with the aid of a machine or device.*"[481] The reasoning of the court on this point cannot be faulted. However, the court emphasised certain parts of this section, which suggests that it considered object code to be protected because it is a fixation in which the work "can be perceived." This is clearly wrong.[482] Object code is seldom intelligible[483] and, as shown above,[484] extremely difficult to read.

By suggesting that object code allows the user to read the work, the court put in place a significant barrier to decompilation. It implies that any copying of the object code must be an infringing action and that, where the user seeks to analyse the work for private study or research, the object code is sufficient. Consequently, there is no need to reverse engineer the work in order to access its underlying ideas.

Furthermore, the reliance on the perception of the work, i.e. a representation comprising text, in order to extend protection to the object code perpetuated the effects of the literary-analogy outlined above. It left no room to argue that any differentiation could be made between the source and object code insofar as the scope of protection

---

[480] *Williams II* at 19.

[481] At 20 (original emphasis by the court).

[482] The technical nature of object code, and the fact that it is not intended to be perceived by humans, is made clear in the sources discussed above at fn 323 and 324. See also the cases in the footnote immediately below.

[483] See *Apple Computer Inc v Franklin Computer Corporation (1982)* at 6 where the court held:

> "In a crude way, object code that has been etched onto the ROM architecture can be "read" by an expert with a microscope and patience. However, the object code in either its binary form or in the silicon chip form is not designed to be read by humans. It is the machine's language."

See also *SAS Institute Inc v S&H Computer Systems Inc* 605 F. Supp 816 (M.D. Tenn. 1985) at 5.

[484] See the discussion in chapter 3 above, in particular the analysis of the program illustration in paragraph 3 1 1 3.

is concerned, because both are represented in perceivable text. As a result, the protection of the text as a whole[485] is again emphasised, and the possibility of distinguishing, between protectable expression and unprotected ideas, further complicated.

4 2 3 4 *Apple v Franklin (1982),*[486] *Apple v Formula International (1983),*[487] *and Apple v Franklin (1983)*[488]

For the purpose of this work, these three cases are discussed simultaneously because: (1) the plaintiff in all three cases is the same party; (2) many of the disputed copyright works are the same; (3) the first and third cases (namely *Apple I* and *Apple III*) are judgments in the same matter while *Apple I* and *Apple II* are both applications for a preliminary injunction; and, (4) the relevant findings of the courts in all three cases dealt with the same issues.[489] Hereinafter the cases are collectively referred to as the *Apple-trio*.

All three cases related to the reproduction and distribution, by the defendants in each case, of the object code of a number of computer programs[490] in which Apple had registered copyright.[491] Two issues were canvassed by the *Apple-trio* that are relevant

[485] This was made explicit by the court in *GCA Corporation v Chance* (1982) 217 U.S.P.Q. (BNA) 718 at 4 where it held:

> "[B]ecause the object code is the encryption of the copyrighted source code, the two are to be treated as one work; therefore, copyright of the source code protects the object code as well."

See also Samuels *Tennessee Law Review* 364.

[486] *Apple Computer Inc v Franklin Computer Corporation* (1982) 545 F Supp 812 (*Apple I*).

[487] *Apple Computer v Formula International Inc.* (1983) 562 F. Supp. 775 (*Apple II*).

[488] *Apple Computer Inc v Franklin Computer Corporation* (1983) 714 F 2d 1240 (*Apple III*).

[489] See the discussion of these issues immediately below in this paragraph.

[490] *Apple I* and *Apple III* concerned fourteen programs, five of which were the subject of the dispute in *Apple II*. At the time of the judgments, the plaintiff held valid copyright registrations in all fourteen programs.

[491] A large part of each judgment also dealt with the contention that a program recorded in ROM or on floppy disks are not sufficiently reduced to material form. This contention was dismissed and is not relevant to the discussion of these cases in relation to the decompilation exception. A further contention, that a distinction must be made between different types of program, to exclude operating systems from protection, was similarly common to all cases and eventually dismissed. An analysis of these particular findings is also outside the scope of this work.

to the legal conception of computer programs and its impact on the development of a decompilation exception.

The first issue relates to the nature of object code and its susceptibility, or not, to copyright protection. The court in *Apple I* states:

> "[It] is not clear that object code, which was not designed to be "read" by a human reader and can only be read by an expert with a microscope and patience, is a language of description. **It cannot teach**."[492]

Despite this, the court is convinced that the automatic nature of compilation is akin to translation of literary work and, therefore, it "preserves the programmer's original force of authorship".[493] Nevertheless, the court raises doubt about the eligibility of object code because it fails the litmus test for protectable expression, namely, a work "directed to a human audience".[494] It also dismissed earlier findings which sought to settle this question with reference to the same wording in the definition of copy in §101 and the definition of originality in §102(a),[495] but does not provide an opinion or make a finding on whether or not object code shall be protectable.

In *Apple II* this issue was not canvassed because both parties conceded that "converting a program from a source code into object code does not deprive the program of copyright protectability if it was previously protectible."[496]

Therefore, the court in *Apple III* revisits this issue extensively and concludes that, precisely because computer programs *are* literary works, the erstwhile requirement that a work should be "designed to be 'read'"[497] by a human is not applicable to

[492] *Apple I* at 28.

[493] At 30.

[494] At 39.

[495] In this respect, the courts in *Williams II* (discussed above) and in *Midway Manufacturing Co. v Artic International, Inc.* held that the wording "perceived, reproduced, or otherwise communicated, either directly of with the aid of a machine or device" is wide enough to encompass object code despite its lack of legibility or communicative value to humans.

[496] *Apple Computer v Formula International Inc* at 9.

[497] *Apple III* at 21.

computer programs.[498] Curiously, the court does not rely on the translation argument to conclude that object code is protectable. Instead, it applies the literary-analogy more directly, with reference to the nature of literary works, and refers to case law[499] in relation to traditional literary works in order to make the point that the act has removed the legibility requirement. It concludes that it was the intention of the legislature that illegible text may qualify for protection and that object code falls within the category of works that can be perceived by a machine.[500] It also confirms the judgment in *Williams II* in this respect.[501]

Important for this discussion, on the development of a decompilation exception, is the cumulative effect of these judgments. It went beyond the argument in *Williams II*, that object code is protectable by virtue of its derivative nature to source code, and declared that object code is, independently, susceptible to copyright protection because it meets the definition of an original literary work, despite its appearance or purpose.

Clearly, this would make it very difficult to argue that the object code should be treated differently to the source code (or any other literary work) when the need arises to reproduce it in order to make the work legible (or more legible, as the case may be). As a result, the *Apple-trio* bolstered the barrier to decompilation created in *Williams II,* by elevating object code to full literary status.

The second issue raised by these cases, which relates to the development of a decompilation exception, is a matter of policy on the balancing of competing interests in protected software.

In *Apple I* the court opines that the plaintiff's works are unlikely to survive a copyright challenge on the merits because, inter alia, injunctive relief would "certainly have a

---

498 *Apple III* at 24.

499 At 21.

500 At 21-2.

501 See paragraph 4 2 3 2 above.

devastating effect"[502] on the business of the defendant. This is because the sole business of Franklin was to make and distribute, along with its Ace 100 computer, unauthorised reproductions of Apple's programs.[503]

This observation echoes the "socially desirable" sentiment of the court in *Synercom* and displays a strikingly similar willingness to overlook reproduction of the programs where it is done for a commercial and competitive purpose. The only difference is that in *Synercom* the court used the idea/expression dichotomy to disqualify the work while *Apple I* opined that it is not a qualifying type of work. But both cases pursue these arguments in order to serve a matter of policy – the competitive interest in leveraging existing computer code.

In *Apple II*, the matter of free competition is raised again[504] and the court provides a more detailed exposition on this issue. After finding that there is no compelling reason in law to distinguish between different types of computer program,[505] it considers whether public policy may dictate a variation in the scope of protection. The court finds that the argument, to permit the free reproduction of Apple's operating programs in the interest of free competition, is not consistent with the evidence in the case. It finds that "[t]here is nothing in this history to suggest that one large firm, such as Apple, is capable of 'dominating' such an expanding market."[506]

The court then defines what it considers to be the correct application of competition policy in relation to copyright in computer programs. It states that "simple economics" suggests that a limitation on the scope of protection, in order to allow reproduction for competitive use, "would hinder, not promote, competition and innovation in the

---

502 *Apple I* at 42 (emphasis added).

503 At 2.

504 Unlike *Apple I,* where the court mero moto introduced the question of policy, in *Apple II* it was raised by the defendant.

505 Computer programs that operate the machine but produce no perceivable result (so-called operating computer programs) and programs with which the user interacts directly (so-called applications computer programs). See *Apple II* at 19-20. See also *Apple III* at 31 where the court makes the exact same finding.

506 *Apple II* at 20-21.

computer market."[507] Therefore, policy dictates that strong protection is necessary to protect the market for new software[508] and avoid profiteering without investment in the research and development of computer programs.[509] The court also confirms that, as a matter of policy, copyright law is the most suitable form of IP for protecting software because all other forms "may inhibit the dissemination of information and restrict competition to a greater extent than copyright".[510]

Consequently, it maintains a proper balance because, on the facts of the instant case, copyright law leaves the defendant free to compete by creating a computer program that performs the exact same function, provided that it does not do so in the "*exact same manner*".[511] All that is prohibited is to copy "the exact number and sequence of bytes or items by which plaintiff's program causes the machine to operate".[512]

This is clearly a reversal of the sentiment expressed in *Apple I*, and the court in *Apple II* acknowledges this fact. It purposefully distinguishes the policy arguments by stating that the facts are different.[513] Nevertheless, *Apple II* clearly expresses a normative judgment on the role of software copyright in promoting innovation through wide protection. The underlying sentiment is not new and widely accepted as the primary economic justification for copyright protection.[514] On this front it cannot be faulted.

Unfortunately, the judgment on policy was made in relation to manifestly infringing activity (direct reproduction for the purpose of distribution) and not in the context of

[507] At 23.

[508] *Apple II* at 23.

[509] At 22.

[510] At 24 with reference to the CONTU report.

[511] At 21 (original emphasis).

[512] At 22.

[513] While the business of the defendant in *Apple I* was reliant for survival on the right to reproduce the programs, the defendant in *Apple II* is not. It does not, however, suggest that the relative position of the defendant in the market or its reliance on copying should play a role in deciding on the scope of the rights in software. In *Apple III* at 46 the court stated categorically that:

> "The size of the infringer should not be determinative of the copyright holder's ability to get prompt judicial redress."

[514] See the discussion in paragraph 2 1 1 above and 5 2 2 2 2 and 5 2 2 3 below.

reproduction for the purpose of learning with a view to creating a competing program. Nor did the court make this distinction. It framed the policy argument so wide that it effectively replaced the sentiments in *Apple I* and *Synercom*. The *Apple II* judgment, therefore, concretised a policy view on the reproduction of code that would find general acceptance as the primary reason not to allow decompilation – a situation that is clearly outside the intended meaning of the court in this case.

In *Apple III* the court also responds to the "devasting impact" statement in *Apple I*, this time in the context of the idea/expression dichotomy and an analysis of the irreparable harm requirement for injunctive relief. The court finds that where an idea for a program is capable of more than one expression, "preservation of the balance between competition and protection" is maintained by the law. It states:

> "The legislative history indicates that §102(b) was intended 'to make clear that the expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of the copyright law'".[515]

This goes some way toward acknowledging that the scope of copyright protection cannot be applied to restrict access to the "processes or methods" of the program. However, it is neither explicit nor implied in this finding that the policy that underpins it is sufficient to permit decompilation. Once again, this is due to the context in which it was made – where the defendant argued competition by reproduction rather than competition by learning.

Furthermore, any value that this finding might have toward justifying decompilation is undone by the court's subsequent statements in relation to irreparable harm. It finds that:

> "[T]he **public interest** underlying the copyright law **requires a presumption of irreparable harm**, as long as there is, as here, adequate evidence of the expenditure of significant time, effort and money directed to the production of the copyrighted material. Otherwise, the rationale for protecting copyright, that of encouraging creativity, would be undermined."[516]

---

[515] *Apple III* at 37-8 with reference to House Judiciary Committee *Copyright Law Revision House Report*.
[516] *Apple III* at 45.

This statement dealt a devastating blow to the development of a general decompilation exception. It emphasises the capital cost of creation as the sole reason why software should be protected against any form of use that would jeopardise the proprietor's "competitive position."[517] It leaves no further room to submit that the potential for harm, in the case of reproduction for the purpose of creating a competing program, is outweighed by public policy.

In other words, the presumption of irreparable harm, read together with the, by now, well established misconception that reverse engineering is a prohibited reproduction or translation, means that any argument for decompilation would have to overcome the hurdle of convincing a court to ignore a presumption of harm. Consequently, it positions public policy firmly in favour of the proprietor and, in addition, obliterates any policy argument in favour of reproduction *regardless* of its purpose. This is the reason why decompilation continues to be viewed, erroneously, as an avenue to wholesale piracy.

4 2 3 5 *SAS Institute v S&H Computer Systems (1985)*[518]

This case is the first to consider, to a limited extent, the reverse engineering of a computer program, by means of informed estimation gleaned from the source code and the operation of the computer. No decompilation occurred in this matter.

At the centre of the dispute was a license, granted by SAS to S&H, to use the source code and accompanying descriptive documentation about the operation of the statistical analysis program SAS79.5. The license agreement stipulated that the program may only be installed on one specific computer, may be duplicated for backup purposes only and may be modified in order to be used. The agreement was exhaustive as far as the rights conferred on S&H were concerned, and specifically prohibited any further reproduction of the program, timesharing use, onward distribution or redistribution of any modified version.[519]

[517] *Apple III* at 43.
[518] *SAS Institute Inc v S&H Computer Systems Inc* 605 F. Supp 816 (M.D. Tenn. 1985) (*SAS Institute*).
[519] *SAS Institute* at 13.

In terms of the license, S&H received the source code for two of the three primary components of SAS79.5, the object code for all three parts and additional material containing descriptions of how the entire program operated and its organisational structure.

S&H obtained the license and the code in order to study the work and make, with the intent to distribute,[520] a version thereof that would be compatible with VAX computers. SAS was not aware of this purpose[521] when it granted the license and was already in the process of converting the SAS79.5 program to be compatible with VAX computers.[522]

S&H installed the program on a different computer at a different location to the terms of the agreement, re-encoded the program by a process of translation, made additional copies of the program, made the source code available to parties other than the licensees, amended the source code extensively and incorporated substantial parts thereof into a new program called INDAS.[523]

SAS objected to these actions and contended that: (1) it would not have licensed the program if it knew about the intention of S&H, which renders the license void and, consequently, deauthorises all actions by S&H in relation to the program; and, (2) S&H's actions exceeded the scope of the license and constitute copyright infringement.

The court relied on extensive expert testimony and had no difficulty in finding that copyright infringement did occur in relation to the highlighted instances of direct copying.[524] Of interest to the current discussion is the court's finding in relation to those examples of similarity that occurred indirectly and not as a result of copying.

---

[520] *SAS Institute* at 10 & 12.

[521] At 11.

[522] At 3-5.

[523] *SAS Institute at* 14-7. According to expert testimony, at least 44 instances (out of 186 000 lines of code in INDAS) were identified as evidence of direct copying.

[524] SAS expert testimony identified 44 instances. Expert testimony for S&H disputed 3 and conceded to 18 instances of similarity. See *SAS Institute* at 17 and 41-2.

The source code provided to S&H did not include the minor “supervisor” portion of the SAS79.5 program.[525] S&H therefore argued that the supervisor portion of INDAS could not have been copied from SAS and must, thus, be excluded from the order restraining it from distributing INDAS.

The court disagrees and finds that, even though the supervisor part of INDAS does not contain a substantial reproduction[526] of the SAS code, it is nevertheless tainted by the extent of reproduction evident in INDAS as a whole. It then makes the following extraordinary statement:

> “The court’s injunction of the supervisor portion of INDAS need not rest upon principles of copyright law.”[527]

Instead, the court states that its decision is based on general principles of contract law. This is not true. The court’s reasoning makes it clear that it was, in fact, influenced by copyright policy considerations couched in contract law.

It states that a general injunction of the whole of INDAS is necessary to restore SAS to the position it would have occupied had S&H complied with the terms of the license and, secondly, to *avoid* unjustified enrichment on the part of S&H.[528] The court considers it a part of the remedy of restitution which, under the circumstances, will prevent S&H from obtaining a benefit from its misuse of the goods it received in terms of the license.[529]

While it is indeed odd to make a pre-emptive finding of unjustified enrichment, particularly where it was not canvassed by the court at all, the reasoning of the court is, perhaps, better understood as a policy decision to protect the “approximately five

---

[525] *SAS Institute* at 13-4. S&H did, however, have access to the object code for this portion as well as the accompanying documentation, but the court made nothing of this fact. It is assumed that there was no evidence of decompilation or that this was never raised by SAS.

[526] The supervisor part of INDAS did contain some instructions copied from the source code of non-supervisor parts of SAS79.5, although these were clearly not considered substantial.

[527] *SAS Institute* at 52.

[528] At 52-4.

[529] At 53-4.

years”[530] of investment in research and development and “18 man-years of labour”[531] expended by SAS.

In this light, it appears that the court simply sought to apprehend the likelihood of future harm to the copyright interest of SAS by protecting the totality of its code from exploitation.

However, on closer analysis this presents a problem. The court ignored the facility to sever the non-infringing work[532] (the S&H supervisor program) from the rest and, instead, chose to ban it based on a spurious application of quasi-copyright (or quasi-contract law) principles. The making of the supervisor program did not amount to copyright infringement. The program was, also, the result of a process of reverse engineering, through study of the surrounding portions of code and the operation of the program. This type of activity was *not* prohibited by the terms of the license, nor was the creation of derivative works *sans* copying.

The only clear basis for this finding is to punish the use of the work where the intention of the infringer was not explicitly authorised. In this case, S&H clearly intended to create a competing program. This, in the opinion of the court, is sufficient reason to find that all expressions created in pursuit of this goal must, somehow, be outlawed.

The court seems to be aware of the vulnerability of its decision because it expends considerable effort to support it, based on the “implied covenant”[533] or “implied duties of good faith and fair dealing”[534] in contract law. With reference to case law that dealt with the distribution of *exact replicas* of dramatic and artistic works,[535] and had no precedential value to this case, the court extends the implied covenant in the context of copyright beyond its recognised scope.

---

530 *SAS Institute* at 4.

531 At 4.

532 At 50.

533 At 35.

534 At 34-5.

535 At 36-8.

Prior to this case, the implied covenant was understood to restrain a licensee from creating a new work based on the idea, theme or title of the work "even if a stranger could create a new work with such idea, theme or title without infringing the grantor's copyright."[536]

Thus, the covenant applies only to a party with a duty to promote the work, not to all licensees.[537] And yet, the court makes it applicable to S&H, who bore no such duty. To justify this leap, the court finds that S&H, by implication, agreed "**as a matter of law** not to use proprietary SAS materials"[538] to produce a new work. By doing so, the court reads an additional restriction into the license agreement in order to support its finding that "S&H's conduct cannot be said to comply with its legal duty of good faith and fair dealing."[539]

This finding is particularly problematic insofar as it significantly undermined the likelihood of a general decompilation exception. The court went to great lengths to prevent S&H from distributing their supervisor program, albeit a small and potentially useless program in isolation, because it was created in order to compete. Neither the fact that the work was non-infringing, nor the fact that it was created without breaching the terms of the license, swayed the court's judgment. Insofar as the supervisor program is concerned, the judgment is baseless.

And yet, its effect is detrimental. The analysis of this case illustrates the flawed logic in anti-decompilation rhetoric: whatever work flows from this step will somehow be improper, even though no reproduction can be espied in the text. This is the reason why decompilation is often closely, and wrongly, associated with adaptation.

Furthermore, the unfortunate extension of the implied covenant created the option to restrain the fair dealing in computer code by way of a copyright license. It makes it possible, and justifiable, to prevent another from analysing a program in order to create

[536] At 35 with reference to *Nimmer on Copyright* 10.11[B].

[537] *SAS Institute* at 36.

[538] At 37-8 (emphasis added).

[539] *SAS Institute* at 37-8.

a competing version, even if the process does not involve any reproduction. Moreover, it is not necessary to make such a prohibition explicit in the license. This means that the law (without regard to copyright law principles) prohibits the use of code as a learning instrument, unless it is proven that the purpose was non-competitive. The burden this place on a defendant is clearly absurd. And yet, it is directly reflected in the limited scope of decompilation exceptions.

4 2 3 6 *EF Johnson Company v Uniden Corporation (1985)*[540]

The judgment in this case goes some way toward curing the defect in *SAS Institute*, insofar as it loosens the restriction on reverse engineering for competitive use. It is also the earliest case under discussion that involved decompilation.

The dispute related to copyright in the computer program that controlled the two-way radio communication devices produced and sold by the plaintiff (EFJ). The LTR program[541] was installed on the mobile radios and repeaters[542] that make up EFJ's LTR communication system. Several versions of the LTR program had received copyright registration. The system is one of a small number of "trunked"[543] mobile radio systems, sold at the time, which is capable of providing uninterrupted radio communication to a larger number of users by efficiently coordinating the use of available radio frequencies and "unutilized airwave spaces."[544] The LTR program is responsible for this coordination effort, called trunked logic. On the repeaters, it receives high-frequency data signals from each mobile radio which identifies the particular device, amplifies the signal and retransmits it to the intended recipient radio. On the mobile radios, the LTR program generates, sends and receives the signals. This allows for two-way communication even where the limited number of licensed radio channels (or radio frequencies) are occupied.

---

[540] *EF Johnson Company v Uniden Corporation of America* (1985) 623 F. Supp. 1485 (*Uniden*).

[541] Entitled the "Grindahl-Barnes EFJ LTR software".

[542] *Uniden* at 9-10.

[543] At 7 & 10.

[544] At 7.

The defendant, Uniden, entered this market by importing and distributing a mobile radio device able to send and receive communications by means of the LTR trunked radio system. In order to be compatible with the LTR system, the Uniden radios contained a computer program that performed the same task as the LTR program. EFJ decompiled[545] the Uniden program, found that it contained a number of similarities to the LTR program and approached the court for a preliminary injunction against Uniden for copyright infringement.[546]

In the process of creating LTR-compatible radios, Uniden carried out a decompilation process[547] on the LTR program similar to the process used by EFJ. In addition, Uniden created flow charts from the disassembled LTR code and used this in creating the Uniden program. The resultant program enabled Uniden mobile radios to work on the LTR system and communicate with LTR devices.

The court applies the iterative test for substantial similarity and, based on expert testimony, focusses on 5 "indicia of substantial similarity."[548] Four of the indicia relate to a specific function of the LTR program each,[549] while the fifth refers to a miscellanea of other similarities.[550]

In relation to the first two indicia, Uniden argues that the similarity is necessary to establish interoperability with the LTR system and are, as such, outside the scope of EFJ's copyright. In the alternative, Uniden submits that these instructions in the Uniden code are derived from public domain work and not reproduced from the decompiled

---

545 EFJ obtained a Uniden mobile radio, copied the code from the storage device, created its own decompilation program and used it to reverse engineer the Uniden program code into assembly language. It then carried out a comparison of the lines of code. See *Uniden* at 11-12.

546 A second allegation, based on trade secret violation, was not pursued at trial.

547 The court provides no detail, only describing the steps taken by Uniden as disassembly of the LTR code. See *Uniden* at 11.

548 *Uniden* at 23.

549 The Barker code (at 23), the H-matrix (at 27), the duplex function (at 29) and the select call prohibit feature (at 31).

550 An arbitrary counter setting in the LTR code reproduced by Uniden, the neat or rambling style of coding in certain parts, duplication of 38 out of 44 subroutines and verbatim reproduction of parts of the LTR user manual in the Uniden manual. See *Uniden* at 33.

LTR code. The court rejects both contentions. While it is true that Barker code and H-matrix coded functions are essential to compatibility, and examples thereof had been published in textbooks,[551] the Uniden code contained versions of these instructions that were identical to the LTR code. In both examples, the LTR code contained unique variations of these functions,[552] constituting protected expressions of these concepts by LTR. The presence of these unique manifestations in the Uniden program served as evidence of direct copying. The LTR code was also not representative of the only[553] or the most efficient method,[554] of achieving the desired outcome, or the only source of this information.

In relation to the other indicia of similarity, the court's general sentiment of a "comity of errors"[555] applies equally to both. The LTR code relating to the duplex function[556] are small remnants of an abandoned attempt to facilitate simultaneous transmission and reception of communications. EFJ never implemented this function and, after abandoning its attempts, removed most of the related code. The three remaining instructions in the LTR code were left in place by accident, and appear verbatim in the Uniden code in the same place, despite the fact that it serves no purpose.[557]

---

[551] *Uniden* at 40-2.

[552] In the case of the Barker code, by limiting the operation thereof to 56 samples, using a specific sampling error table bespoke to the Intel processor used by LTR (and not used by Uniden devices) and applying an error threshold of 8 or fewer for synchronization. See *Uniden* at 40-1. In the case of the H-matrix, by loading the instructions in reverse order and applying an inverse version of the matrix despite this not being necessary to enable a device to work on the LTR system. See *Uniden* at 42.

[553] At least 32 variations of the H-matrix were possible.

[554] In fact, evidence showed that the Uniden device was capable of more efficient functioning than the LTR devices, but was restricted by the fact that it employed coded instructions copied from the LTR program. See *Uniden* at 25 and 27.

[555] *Uniden* at 31.

[556] Which would allow the user to hear communications while transmitting a communication at the same time.

[557] *Uniden* at 29-30. Uniden argued that these instructions were included for future use, in the event that it decides to add a duplex function. The court was not convinced that the three identical lines of code were, under these circumstances, independently created by Uniden for this purpose.

The select call prohibit feature in the LTR code contained an inadvertent error which generated a false overload alert (or "busy" signal)[558] of the system and disallowed communications under these circumstances.[559] EFJ later amended the code to address this problem, but the same error was present in the Uniden program. The court found it unlikely that Uniden inadvertently made an identical mistake[560] and, with reference to *Williams II*, consider this "presence of identical errors in copyrighted and infringing computer programs"[561] as further evidence of copying.

In light of these facts, the court finds that the Uniden program is substantially similar to the LTR program. It then moves to consider whether Uniden should be free to reproduce these parts of the LTR program because: (1) it is not subject to protection by virtue of its commonality in the industry (non-originality); or, (2) it is unprotectable ideas.

The court dispensed with both counterarguments. It finds that EFJ "has contributed something 'recognizeably its own' to the compilation of standardized programming techniques and innovative EFJ-generated instructions"[562] by implementing the Barker code and H-matrix instructions in a unique manner. In addition, with reference to *Synercom,* the court agrees that a unique combination of unoriginal component parts, such as the instant case, may be sufficient to constitute an original and copyrightable work.[563]

Regarding the second counterargument, the court finds that Uniden did not reproduce the underlying ideas necessary to achieve interoperability (a situation which would have avoided any copyright liability), but reproduced the exact expression of these ideas from EFJ's code.

---

[558] *Uniden* at 31.

[559] At 31.

[560] At 32.

[561] At 32.

[562] At 42.

[563] At 43.

The court frames the test for the separation of idea and expression in software copyright cases, with reference to *Apple I* and *Apple II* as follows:

> "Whether other programs can be written which perform the same function as the copyrighted program. If other programs can be written or created which perform the same function as the copyrighted program, then that program is an expression of the idea and hence copyrightable. If a specific program, even if previously copyrighted, is **the only and essential means of accomplishing a given task**, their later use by another does not amount to infringement."[564]

In light of the fact that the LTR program was not the only way of achieving trunked radio communications on the LTR system, or the only system of this kind on the market, the court finds "the record amply demonstrates that an LTR-compatible software program could have been written without verbatim duplication of EFJ's version 3.0 program".[565]

It concludes:

> "Defendant was required to copy plaintiff's Barker word, as discussed above. Virtually all other aspects of defendant's program could have been independently created, however, **without violence to defendant's compatibility objective**. Defendant has reproduced the expression, not merely the idea of plaintiff's copyrighted program."[566]

This judgment is remarkable for its clarity on the role and purpose of decompilation and the impact thereof on the idea/expression dichotomy. It is clear that decompilation of protected source code is not, *per se*, an act of infringement.

In fact, the court makes it expressly clear:

> "The mere fact that defendant's engineers dumped, flow charted, and analyzed plaintiff's code **does not, in and of itself, establish pirating**. As both parties' witnesses admitted, dumping and analyzing competitors' codes **is a standard practice in the industry**."[567]

---

[564] *Uniden* at 48-9 (emphasis added).

[565] At 50-1.

[566] At 52.

[567] *Uniden* at fn17 (emphasis added).

The term “dump” is common vernacular in the software industry to describe the act of copying or transferring computer instructions *en masse*. The *Oxford Advanced Learner’s Dictionary* define “dump” in relation to computing as: “to copy information and move it somewhere to store it”.[568] The act of analysing is only possible after re-encoding, which means that decompilation by so-called translation, as described in this judgment, is clearly a part of the justified acts listed by the court here. Even if this is not the case, the meaning of “dump” is clearly a form of reproduction, considering that data cannot be physically moved but must be replicated in order to shift it from one storage medium or device to another. Therefore, the act of decompilation, insofar as it is said to be an act of reproduction, is included in the ambit of this finding.

Furthermore, where the act of decompilation is conducted in pursuit of interoperability or cross-compatibility, even for direct competitive purposes, copyright law still does not apply. Only in the event where reproduction of the post-decompiled code occurs, in a manner that is both substantially similar and related to a protected expression, rather than an underlying idea, would copyright law find application.

This draws a neat threshold in decompilation cases, beyond which infringement may occur: reproduction of original code into a new work. The act of decompilation is, therefore, neither indicative of copying nor, to any degree, a prohibited form of reproduction in terms of the principles of copyright law.

The accuracy of this construction is supported by the analysis of policy considerations conducted by the court. It considers the competitive interest of parties such as Uniden and confirm, expressly, that these steps are justified in the interest of fair competition. However, the threshold is passed where a party engages in decompilation and thereafter applies the results in such a manner that it gains an unfair competitive advantage. This would be the case where the decompilation results were used to create a new program by copying the code, instead of independent effort at creating a program with the same or similar abilities.[569]

[568] Oxford University Press *Oxford Advanced Learner’s Dictionary* 7ed (2006) 456.

[569] *Uniden* at 55.

4 2 3 7 *Whelan Associates v Jaslow Dental Laboratory (1986)*[570]

This case, along with the judgment of the court a quo[571] and the *SAS Institute*[572] decision, established copyright protection in the non-literal elements of computer programs and applied it to the structure and organisation of software and, to a lesser extent, the visual output[573] of the program. This marked a significant development in copyright law regarding, in particular, the interpretation of the idea/expression dichotomy.

Of interest to the current analysis are the reasons why the court saw fit to expand the scope of protection, because: (1) it directly affected the ability of programmers to create a new work that performs the same function according to the balancing principle espoused in the *Apple-trio*; and, (2) substantially altered the test for infringement to the extent that, when comparing software, a value judgment must be made on the importance of the structural arrangement which, inevitably, makes it more difficult to argue that decompilation should be permissible in order to discover the logical and syntactical expressions.

In this case, unlike any of the previous matters discussed above, there was no direct reproduction of code. The two works were also expressed in different programming

---

[570] *Whelan Associates Inc v Jaslow Dental Laboratory Inc* 797 F.2d 1222 (3d Cir. 1986) (*Whelan II*).

[571] *Whelan Associates v Jaslow Dental Laboratory Inc (1985)* (*Whelan I)*.

[572] *SAS Institute* at 41. The court did not deal with non-literal elements in detail, but laid the foundation for it and emphasised its value to the proprietor (at 31-2). It referred to similarities in the structural arrangement of the programs (at 22) as one of many indicators of copying. It also stated, with reference to authority, that:

> "Substantial similarity, of course, does not require literal identity; 'a play may be pirated without using the dialogue.'"

[573] The development of protection for visual output as a non-literal element of computer programs is complicated by the analogous protection previously extended to this type of expression as audio-visual works. Both *Whelan* cases also dealt with visual output. However, this is not relevant to this work insofar as it has no impact on the development of a decompilation prohibition. Visual output can easily be discovered without any effort at decompilation.

languages.[574] The original work, DENTALAB, was owned by Whelan[575] and used, and distributed under license, by Jaslow. The contentious work, DENTCOM, was developed and owned by Jaslow[576] to perform the exact same functions. The intention of Jaslow was to create a program that would perform the same functions as DENTALAB, but also be compatible with a wider range of types of computer.[577] This is the reason why the two programs were written in different programming languages.

Jaslow achieved this intended assimilation of function by, inter alia, employing the same logical arrangement in the structure of the program itself and in the structure of its constitutive parts, with minor variations to the order in which these steps are taken by the computer. In practice, and in the judgments, the constitutive parts, each of which have an identifiable and specific task, are referred to as *sub-routines*: a term used to describe a set of instructions that make up a computer program and perform a specific task when instructed to do so by another part of the container program (sometimes referred to as a software package).

In this case, for example, the DENTALAB computer program contained a computer program (or sub-routine) called MOEND. Its purpose, when activated, was to initiate a number of other sub-routines in a specified order.[578] The sub-routines in this case

[574] DENTALAB was written in EDI, DENTCOM was written in BASIC.

[575] Copyright in this work vested, originally, in another entity (Strohl) and was transferred to the original programmer in her capacity as a member of the plaintiff Whelan.

[576] A portion of the work was created by Jaslow and another part by a third party (Jonathan Novak) working under an employment agreement.

[577] This intention is clearly established on the facts. See *Whelan II* at 6-9.

[578] The process is described by the court in *Whelan II*, at 85-86, with reference to the transcript of testimony in *Whelan I*, as follows:

> "The Dentcom system, it first prints product group report, and then prints the monthly customer sales analysis. In Dentalab, just reverses, prints sales analysis first, product group second. After that, both systems do the same thing in the same order. They now do accounts receivable aging, since a month has gone by they have to update all the 30 days, 60 days, et cetera, calculate service charges. Then they print the monthly AR reports that had to do with service charges, only those that involve service charges, they both do that. Then they both print the age file balance, balance report, and following that they print the month and accounts receivable report. That's the total accounts receivable report [sic]. Then they both go through and look for accounts that are not active that month, and print a list of these accounts, accounts

each performed a rudimentary and standard task, such as to collect information from a data file that meet certain criteria (as a list of customers with outstanding balances) and display it on the screen, or perform an accounting task (calculate a balance of payments).

The court finds that, based on expert testimony, a substantial similarity exists between the structure of both programs. The court did not itself conduct an analysis to arrive at this conclusion. It rejected the testimony of Dr Hess, Jaslow's expert, preferring the testimony of Dr Moore, Whelan's expert, because the court *a quo* found that his analysis of the program *structure* was more thorough and applicable to the case.

Dr Hess, who examined the source and object codes, found that the second work was not "directly derived" from Whelan's program and that sufficient dissimilarity existed to avoid a claim of infringement.[579] The court, however, found fault with this analysis.[580]

It chose to rely on Dr Moore's opinion. He examined the file system (an ordinary arrangement of information) and only five sub-routines,[581] which represented a fraction of the programs.[582] He concluded that the file systems were identical while the similarities in the arrangement of the sub-routines of DENTCOM suggest that:

> "[T]he person who wrote the IBM-PC Dentcom system had to have access to the plaintiff's Dentalab source code **to be able to understand the system** and to be able to **use all the same sequential operations**."[583]

This factor, namely, *access* to the source code, is the first aspect of these cases that has a direct impact on the development of a decompilation exception. The court found

---

> not serviced, an account that doesn't have any access. The final thing that the Dentcom system does is to calculate the new AR total for the entire lab, which I mentioned is contained in the company file."

579 *Whelan I* at 24 and 43.

580 *Whelan II* at 82. There is also a suggestion, which is not explained, that Dr Hess's examination of the code dealt with a version of code that was based on a coding system that was not employed by Jaslow. See *Whelan II* at 23-4 and 43.

581 The function of these programs is described as: "order entry, invoicing, accounts receivable, end of day procedure, and end of month procedure." *Whelan II* at 14.

582 *Whelan II* 14 & 76-7.

583 *Whelan I* at 44.

that infringement was established because the court a quo had sufficient evidence of copying, namely "access to and use of the plaintiff's source code, the similarity in the structure of the programs and in the flow charts."[584] However, this finding is based on substantive mistakes some of which seem intentional.

Firstly, the court acknowledged that "for certain tasks there are only a very limited number of file structures available, and in such cases the structures might not be copyrightable and similarity of file structures might not be strongly probative of similarity of the program as a whole."[585] And yet, it seems content to make a finding of similarity based on Dr Moore's testimony which was, in no small part, influenced by the similarity of file structure. Furthermore, file structure is not an overly complex or technical expression such as computer code. There is no reason why the court had to rely on the testimony of a non-jurist, instead of making its own comparison, in order to make a finding about whether or not the file structure was copied.[586]

Second, and most important, the court is persuaded that reproduction is the cause of the similarity in program structure because Jaslow had access to the source code and, therefore, *could have used* it to create a similar structure. In fact, the court agrees with Whelan that the similarities in file structure, program structure and visual appearance means that Jaslow not only *could have used* the source code as inspiration but *did use* it. On this basis, the court makes its finding of infringing reproduction.

At first blush, this is not extraordinary. To infer reproduction from the combined factors of access to the infringed work (the so-called causal nexus) and substantial similarity, is standard practice in copyright law. However, the manner in which the court applied it in this case leads to a highly problematic conclusion: it is an act of infringement to mimic the appearance or structure of a program if it is based on a perusal of the code.

[584] *Whelan II* at fn45.

[585] At fn43.

[586] In fact, the court misapplied the iterative approach to the substantial similarity test, which is based on expert testimony under circumstances where a direct comparison is impossible or a general impression of the work cannot be formed (such as computer code written in different languages), instead of applying the traditional ordinary observer test. See *Uniden* at 21-2 where the intended application of the iterative approach is explained.

This conclusion is inescapable. If, as in this case, the degree of similarity indicates substantial copying, and there was access to the code, infringement must follow. Consequently, it does not matter that there may have been only one or a limited number of ways in which to structure the program as *efficiently* as possible – a factor the court acknowledged,[587] but chose to ignore. It seems that the court was only willing to accept that the similarity in structure may be excused if it is *essential* to the function of the program.[588] If not, as long as the structure that is duplicated is not the sole possible structure,[589] it would be infringement if a perusal of the code leads to a substantially similar structure.

By this logic, the decision to extend protection to non-literal elements regarding structure means that the code is deserving of protection in relation to its text *and its message* in the same way that other types of literary work are protected against more than just verbatim reproduction. The court explains that, because "one can violate the copyright of a play or book by copying its plot or plot devices,"[590] the law clearly intended that computer programs should benefit from protection outside the limit of textual reproduction.

Regardless of the merit, or demerit, of extending protection to program structure, these cases introduced a significant error in thinking about computer programs and copyright law insofar as decompilation is concerned: access to source code will betray the secret efficiencies which give that program its competitive advantage. Thus "one can approximate a program and thereby gain a significant advantage over competitors even though additional work is [or may be] needed to complete the program."[591]

In this light, it is easy to see why the law was quick, when called upon to do so, to consider decompilation a threatening activity: if source code is so vulnerable to

[587] *Whelan II* at 20-1.

[588] At 52.

[589] Scott traces this sentiment to the judgment in *SAS Institute* at 825. See Scott *Scott on Information Technology Law* 2-100.

[590] *Whelan II* at 35.

[591] At 48-9.

exploitation, the additional layer of protection, offered by encryption to object code, may not be eroded by copyright law.

4 2 3 8 *Atari Games v Nintendo* (1991),[592] *Atari Games v Nintendo* (1992)[593] and *Atari Games v Nintendo* (1993)[594]

The series of judgments in this case, hereinafter collective the *Atari-trio*, along with the *Sega* cases,[595] established a decompilation exception, by way of the fair use doctrine, in American law.

The facts of this case are well known and frequently detailed in jurisprudence.[596] For the purpose of this work, a summation of the salient points will suffice.

For the purpose of securing exclusive compatibility between the NES games console and game cartridges created by, or under license from, Nintendo, the computer program 10NES was installed on both the console and the game cartridges to act as a security device guarding against the use of unlicensed game cartridges in the NES console. Nintendo's copyright in the 10NES system was not disputed[597] and a copy of the program had been deposited with the Copyright Office upon registration.[598]

---

592 *Atari Games Corporation and Tengen Inc v Nintendo of America Inc and Nintendo Company Ltd* (1991) U.S. Dist. LEXIS 5519 (*Atari I).*

593 *Atari Games Corporation and Tengen Inc v Nintendo of America Inc and Nintendo Company Ltd* (1992) 975 F.2d 832 (*Atari II).*

594 *Atari Games Corporation and Tengen Inc v Nintendo of America Inc and Nintendo Company Ltd* (1993) 1993 U.S. Dist. LEXIS 8183 (*Atari III)*.

595 See paragraph 4 3 2 9 below.

596 See the following for a comprehensive description of the facts: Cohen J E "Reverse Engineering and The Rise of Electronic Vigilantism: Intellectual Property Implications of 'Lock-Out' Programs" 1995 *Southern California Law Review* 68 1091 1101-3; Dallas S E "Computer Copyright Protection Narrows as Video Game Giants Battle in Atari v Nintendo" 1994 *Denver University Law Review* 71 739 748-750; Rice D A "Sega and Beyond: A Beacon for Fair Use Analysis…At Least As Far As It Goes" 1994 *University of Dayton Law Review* 19 1131 1149-1152.

597 *Atari I* at 7.

598 *Atari II* at 18. Atari did not attempt to rebut the presumption of originality conferred on the 10NES program by registration.

The 10NES program maintained this closed system by matching the unique data signal produced by the program on the cartridge, generated by the slave or key chip,[599] with the outcome expected by the program in the console, generated by the master or lock chip.[600] If the master/lock program receives a signal from the slave/key chip that matches the data signal it expects from an authorised cartridge, the console will execute the game instructions stored on the cartridge. If it receives no data signal, or a signal that was not produced by the 10NES program, the game on the cartridge cannot be played on the NES console.[601]

To enable the games and game cartridges, produced by Atari, to be accepted by a NES console, Atari created the RABBIT computer program to perform the same unlocking (key) function as the 10NES program. In other words, the RABBIT program was created to ensure *compatibility* between Atari cartridges and the Nintendo console. Similarly, it sought to secure *interoperability* between the RABBIT program and the 10NES program by aligning its output (the key and lock data signals) correctly.

The ensuing copyright dispute about these programs rested on two broad contentions: (1) literal copying of the 10NES code which occurred during decompilation; or (2) substantial similarities in the functions, which went beyond copying of the ideas, constituting reproduction of protected expressions.

The process by which the RABBIT program was created is important to this analysis. In order to understand the operation of the 10NES program and replicate the key

[599] The term chip is used to describe a microprocessor which executes the 10NES program in this case.

[600] *Atari II* at 3.

[601] The technical operation of the data stream is described by the court in *Atari III* as follows:

> "At a fundamental level, the signal stream involves the technical process of data communication between two microprocessors. In order for any communication to occur, the output pin on the cartridge chip must be connected to the input pin on the console chip. When the output pin is set to a high voltage, the console chip will read the value on its input pin as a "1" and when the output pin is set to low voltage, the console chip will read the value "0." Timing between the two chips is also a key factor in data communication. Communication occurs only when the console chip is "expecting" to receive data and "looking" at the value on its input pin. Whether the output pin is set high or low matters only at those moments in time when the console chip is reading a value from its input pin. From the perspective of the console chip, the value on the cartridge's output pin is irrelevant at all other times." *Atari III* at 9-10.

function thereof, Atari accessed, by way of microscopic examination of the storage chips, the object code of the program and painstakingly copied it by hand.[602] The first attempt at this process failed to yield a useful result, presumably because Atari made mistakes while transcribing the series of 1s and 0s. Atari made a second attempt to "deprocess"[603] the chip and transcribe the object code. Thereafter, it unlawfully obtained[604] a copy of the 10NES code from the Copyright Office. This allowed it to "correct errors in this transcription."[605] Because the resultant transcript of "ones and zeros convey little, if any, information to the normal unaided observer,"[606] Atari then "entered [by hand] this copied 10NES object code into a computer which aided in understanding the ideas in the program."[607] This is the process of decompilation, as described by the court, that is the focus of the current analysis.

The process created an approximation of the 10NES source code, which could be read and understood. Aided by this, Atari "developed its own program"[608] called RABBIT, written in a different programming language and according to the restrictions imposed by the use of a different microprocessor.[609]

Clearly, the decompilation steps taken by Atari did not involve the making of an unauthorised reproduction or translation of the 10NES program. It fits squarely within the scope of permissible use outlined by the court in *Uniden*.[610]

---

602 *Atari II* at 5.

603 At 2.

604 Atari misrepresented the state of pending litigation between it and Nintendo in order to qualify for access to the record in terms of the Copyright Office Regulations. At the time, "no controversy at all existed when Atari acquired the 10NES program from the Copyright Office." *Atari II* at 24. Consequently, the copy of the program obtained by Atari was an unauthorized reproduction of the work in terms of section 106(1) of the US Copyright Act.

605 *Atari II* at 6 and *Atari I* at 5.

606 *Atari II* at 32.

607 At 25 and 32.

608 *Atari I* at 6.

609 *Atari II* at 6.

610 *Uniden at* fn17. See the discussion in paragraph 4 2 3 6 above.

However, in *Atari I,* the court suggested that this process amounted to prohibited conduct in the form of "intermediate copying,"[611] based on the finding in *SAS Institute*. In the latter case, intermediate copying was deemed infringing only because it related to reproductions of the program in a manner that was unlicensed[612] – constituting an infringing copy *ab initio*. All subsequent or intermediate copies were therefore infringing copies, whether they could be said to be intermediate or not.

The facts in *Atari I* differ significantly. Some of the intermediate copies made by Atari, namely, the transcripts of object code, did not flow from an infringing reproduction of 10NES code. To this extent, the court's argument is incorrect and its reliance on *SAS Institute* misplaced.

The only intermediate copy that was an unauthorised reproduction is the version of 10NES code unlawfully obtained from the Copyright Office. Insofar as the finding in *Atari I* relate to the making of that copy, it is correct. Insofar as it suggests that the process of decompilation *per se* involves the making of infringing intermediate copies, it is wrong.

However, in *Atari II*, the court points out that the unlawfully obtained copy was used to make alterations to the extracted object code before it was subjected to the decompilation process.[613] This effectively tainted all subsequent reproductions of the work by a process of "derivative copying."[614]

Therefore, insofar as intermediate copying is concerned, *Atari I*, read with *Atari II*, determined that it amounted to infringement because Atari used an unauthorised copy of 10NES. The relevant copy in this context is the unlawfully obtained code from the Copyright Office. It does not mean that the process of de-proccesing the chip to access the object code, transcription thereof, the process of decompilation that followed or the creation of the source code involved the making of an infringing intermediate copy

---

[611] *Atari I* at 14.

[612] See *SAS Institute* at 38-9 and paragraph 4 2 3 5 above.

[613] *Atari II* at 31.

[614] At 31.

per se. Had it not been for the intervening act of infringement by obtaining an unauthorised copy *and* the use thereof in preparing for decompilation, none of the above actions by Atari would have attracted liability based on intermediate copying.[615]

This reasoning is fundamental to the development of the fair use exception in this case. It successfully separates the process of decompilation from the making of an infringing copy. This allowed the court to view the process in isolation, free from the tortuous application of copyright law, and the unwarranted connotations to piracy, established in previous case law.

In *Atari II* the court then outlines the decompilation process on the facts and conclude that it amounts to fair use, based on the following principles:

(1) the purpose of copyright protection, as described by the IP clause in the Constitution,[616] is to promote the progress of science and "not the rewarding of authors";[617]

(2) society has an interest in the free flow of ideas and information;[618]

(3) the Copyright Act "encourages others to build freely upon the ideas and information conveyed by a work;"[619]

(4) the rights of authors mentioned in the Copyright Act are subject to limitation;[620]

(5) "the author does not acquire exclusive rights to a literary work in its entirety"[621] but subject to the limitation that "society is free to exploit facts, ideas,

[615] *Atari II* at 31.

[616] United States Constitution article 8 §8(8) which reads: "[…] to promote the progress of science and useful arts, by securing for limited times to authors and inventors the exclusive right to their respective writings and discoveries." The finding here echoes the statement of the supreme court in *Feist Publications Inc v Rural Telephone Service Company Inc* (1991) 111 S. Ct. 1282 at 1290 that it is a constitutional imperative that the copyright act should promote the progress of science rather than "reward the labor of authors".

[617] *Atari II* at 26.

[618] At 26 with reference to *Sega.*

[619] At 26.

[620] *Atari II* at 27.

[621] At 27.

> processes, or methods of operation in a copyrighted work";[622]
> (6) the Copyright Act permits anyone "in rightful possession of a copy of a work to **undertake necessary efforts to understand** the work's ideas, processes, and methods of operation."[623]

For these reasons, the court finds that it will amount to fair use within the meaning of section 107 to reverse engineer a computer program in order to "discern the unprotectable ideas"[624] and make use of, or build upon, the advances made in that work.[625] This is the decompilation exception.

In addition, to eliminate the doubt about intermediate copying and decompilation, the court states:

> "An author cannot acquire patent-like protection by putting an idea, process, or method of operation **in an unintelligible format [such as object code] and asserting copyright** infringement against those who try to understand that idea, process, or method of operation."[626]

It makes it clear that the development of the fair use doctrine is justified by the legislative history of section 107 to adapt to new technological innovations[627] and that, where the nature of the work requires certain steps to be taken in order to "understand the ideas and processes in a copyrighted work",[628] the nature of the work itself supports the fair use.[629] The court refers to intermediate copying specifically in this context as an example of steps that would be necessary, by virtue of the nature of the work, to access the teaching of the work. This means that, even if the court made a

[622] At 27.

[623] At 27 (emphasis added).

[624] At 30.

[625] At 29.

[626] At 27 (emphasis added).

[627] At 38 with reference to the House Judiciary Committee *Copyright Law Revision House Report* at 66.

[628] At 29.

[629] *Atari II* at 29.

mistake in its earlier finding that separated decompilation and intermediate copying, it finds here that intermediate copying is itself a form of fair use.[630]

It is important to note that this is a twofold exception for reverse engineering. In other words, in the first instance, any steps necessary to discover the ideas, structure, meaning or any other part of the program by any process is fair use. In the second instance, if the process required, or involved, the making of an intermediate copy, that copy is fairly made and will not render the process unfair. Thus, the fairness of the exception does not stem from, nor it is dependent upon, the declaration that intermediate copying during reverse engineer is fair. In fact, the opposite is true. The process is declared fair use, which means that any reproduction that occurs during this process will be non-infringing.

Consequently, a reverse-engineering process that does not involve intermediate copying will, nevertheless, still be fair use. Similarly, an intermediate copy made during reverse engineering is fair twice over: by virtue of the fair process that made it and the fair purpose for which it was made.

To make this clear, the court states that *Atari I* was wrong[631] when it "assumed that reverse engineering (intermediate copying) was copyright infringement"[632] and that "Atari's efforts to reverse engineer the 10NES chip to learn the ideas in the program will not alone support a copyright infringement claim."[633]

Despite this, the court finds that the decompilation process carried out by Atari did not qualify as fair use.[634] Because Atari used an unauthorised copy of the 10NES program in this process, it would violate the "'good faith' and 'fair dealings' underpinnings of fair use"[635] to invoke it under the circumstances.

---

630 At 29.

631 At 33.

632 At 33.

633 At 38.

634 At 32.

635 *Atari II* at 32 quoting from *Harper & Row Publishers Inc v Nation Enterprises* (1985) 471 U.S. 539.

In addition, the RABBIT program contained substantial reproductions of the 10NES code that, in the opinion of the court, indicated reproduction of the protected expressions.[636] In particular, RABBIT contained instructions similar to the master/lock portion of 10NES, which was entirely superfluous to the purpose of interoperability.[637] The RABBIT program also generated a data stream *identical* to the 10NES program despite the fact that a "multitude of different ways to generate a data stream which unlocks the NES console"[638] existed and only a small portion of the data stream was necessary to make the cartridge compatible with the NES system.[639] Atari contended that, although it reproduced more than was necessary to achieve interoperability at the time,[640] all of these similarities were necessary to ensure *future* interoperability[641] in the event that Nintendo should change the 10NES program. *Atari I* did not consider this argument and merely declined to accept the theory that future interoperability could be raised as a defence to copying.[642] Apart from a passing reference thereto, *Atari II* did not entertain the contention at all.[643]

As a result, the court in *Atari II* upheld the preliminary injunction granted in *Atari I* and found that it was correct when it "detected similarities between the programs beyond the similarities necessary to accommodate the programming environment, or similarities necessary to embody the unprotectable idea, process, or method of the 10NES program."[644]

This placed an important limitation on the scope of the fair dealing exception: the right to decompile includes the right to make copies of the work *for that purpose* only,

---

636 *Atari II* at 37.

637 The master/lock portion is only used by the NES console. In order to make the Atari cartridges work with the NES console, Atari only needed to replicate the slave/key portion's functionality. The additional master/lock code in RABBIT had no function.

638 *Atari II* at 18-9. Expert testimony in *Atari I* stated that "billions" of suitable possibilities existed. See *Atari I* at 6.

639 *Atari I* at 5.

640 At 7.

641 *Atari I* at 5 – 7 and *Atari II* at 37.

642 *Atari I* at 7.

643 *Atari II* at 37.

644 *Atari II* at 35 and 42-3.

namely to discover ideas etc. The justification for this exception rests in this limitation. The work, once decompiled, is intended for use to stimulate the creation of new work. Therefore, the results may only be used to copy the unprotected elements discovered by the process of decompilation. Any copying of the expressions embodied by the original author, insofar as it was revealed by decompilation, may not be reproduced. In short, after decompilation, it is expected that the user will write new code and refrain from piracy of the protected elements, namely, the literal code or the protected expression conveyed by the code.

The fair use exception is, therefore, *not* a right to copy anything. The court makes it clear that fairness refers to the method of use, not the making of reproductions.[645] Consequently, the decompilation exception may not be used to justify any contentious reproduction in a derivative work.[646] Similarly, the act of decompilation may not be invoked as an indication of intent to commit copyright infringement.

This has effectively amended the test for substantial similarity, insofar as it relates to the first element of proven access to the work. No longer is it possible to argue that the similarities in code *are more* significant merely because it can be shown that the work had been decompiled, granting access to the code. In other words, factual access to the code remains an element of the test, but it must be applied consistently with other types of work which could, by their nature, be read and understood without decompilation.

In the light of this, the fair use exception is a right to gain access by whatever technical means, even if it includes intermittent copying. It is not, and should never be interpreted as, a right to make reproductions or facilitate the making of reproductions. Thus, the decompilation exception is not intended to circumscribe, or in any way influence, the extent of permitted reproduction from source or object code.

---

[645] At 38.

[646] The court clearly states, at 38:

> "Atari's efforts to reverse engineer the 10NES chip to learn the ideas in the program will not alone support a copyright infringement claim."

Unfortunately, from the outset, the courts made mistakes in the application of the decompilation exception, which, erroneously, blended the decompilation exception with the idea/expression separation. It failed to understand the purpose and scope of the exception and, instead, used it to influence decisions on whether or not a particular similarity is permissible or not. In other words, it attempted to determine whether a particular instance of similarity (reproduction of a protected expression) may be justified because it was written based on decompiled code.

A clear example of this misapplication is found in *Atari III*.[647] In this case Atari brought a motion for summary judgment based on, inter alia, the same contention regarding future compatibility, between the Atari cartridges and the NES system, it had raised in *Atari I* and *Atari II.* The difference,[648] in respect of copyright law, is that in this case Atari submitted that reproduction of the data stream and, by implication those portions of code that generate the totality of the data stream, is justified because it does not constitute copyrightable subject matter[649] (i.e. it is not a suitable type of work or it does not meet the requirements for protection).

While this is a distinction in theory, as the court makes clear, it is a practical red herring.[650] If the court now finds in favour of Atari on this point, it would be entitled to emulate the full data stream and all of the code that generate that specific data stream. If the court in *Atari II* found in Atari's favour, it would also have excused the emulation of the data stream and all of the associated code. Thus, the outcome is the same – it determines the extent to which Atari is entitled to copy. The legal basis is also the same – in both situations a positive finding for Atari would be based on the idea/expression dichotomy (a determination that the code that generates the data stream is unprotectable subject matter).

[647] See also the analysis of this decision by AR Miller "Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?" (1993) 106 *Harvard Law Review* 977 at 1018, where the author states that the court's reasoning was "not fully persuasive".

[648] In *Atari I* and *Atari II*, the contention of future compatibility was raised as a defence to infringement to justify the extent to which Atari reproduced code. In Atari III, it is not raised as a defence but, instead, as the basis for a declarator regarding the scope of copyright in 10NES.

[649] *Atari III* at 6.

[650] *Atari III* at 20.

The court first examines the nature of the data stream itself and finds that it does not qualify for protection. It correctly distinguishes the data stream as output, or mere data produced by the 10NES code, from other data constituting computer instructions.[651] Therefore, it may not qualify for protection as part of the computer program and must satisfy the requirements for protection as a distinct work.[652]

The data stream may potentially be protected as a compilation consisting solely of a series of numbers in a predetermined order if it reflects the necessary originality in the selection, arrangement and coordination of information.[653] On the facts, the court finds that it lacks originality, as defined by the court in *Feist,*[654] because the selection and arrangement of numbers in the data stream are *arbitrarily* determined based on a random value issued by the program code.[655] Therefore, the data stream, although unique, is not original insofar as it does not embody the intellectual endeavours of Nintendo. The fact that the uniqueness of the data stream is a direct consequence of the protected, clearly original, effort of the 10NES programmer, does not affect the originality of the data stream.[656] Consequently, the court finds that the data stream does not qualify for protection.[657]

At this point, the analysis in *Atari III* is non-contentious and the enquiry should have ended here. However, the court then embarks on a tangential examination of the message conveyed by certain parts of the data stream. It focusses on the silences between constituent parts of the whole of the data stream which separate one series

[651] *Atari III* at 7, 11 and 16.

[652] At 14.

[653] At 14.

[654] *Feist Publications Inc v Rural Telephone Service Company Inc*.

[655] *Atari III* at 14.

[656] For the sake of clarity, the relevant portion of the judgment (at 14) reads as follows:

> "Nintendo's signal stream consists of numbers which are predetermined by the random "seed" value generated at the beginning of the 10NES program. These numbers work as a "lock" precisely because they are arbitrary and cannot be determined prior to execution of the program; they are not the result of specific "choices as to selection and arrangement" and therefore do not meet Feist's originality requirement." *Atari III* at 14.

[657] *Atari III* at 15 and fn13.

of numbers from the next. The court considers this important because of the way the 10NES lock and key mechanism works.

In short, the 10NES program generates a random number (the seed number)[658] and uses that number as the basis for calculating a series of other numbers separated by periods of silence represented by 0s. These numbers, collectively, comprise the lock data stream, which is sometimes referred to in all of the judgments as a "song."[659] This process is carried out by the console according to a predetermined method for computation set by that part of the 10NES program that generates the data stream.[660] All compatible cartridges also generate a series of numbers with silences, without the aid of the seed number but according to the same calculation method dictated by the 10NES program. This means that the numbers generated by the cartridge will be predictable and fall within the range of numbers that could only be created by the 10NES program (or a similar set of instructions). In this way, the data streams "match", although they are not duplicates.

The exact functioning of the 10NES in relation to the data stream was redacted before publication of *Atari III*. However, a simple mathematical example to illustrate the technique would suffice:

10NES "MASTER/LOCK" process on console:

1. *Seed number is 5.*
2. *Multiply seed number by 10, shift all characters UP by the number of times this step has been carried out, add the numbers 131 at the end. Wait 1ms.*
3. *Repeat process in step c three times.*

The calculation would thus be:

- (5 x 10) = 50
- 50 becomes 61 (shifted up one for first iteration). [This process accepts that the number range of 1 to 9 is used in both directions. If 9 is shifted up by one it restarts the sequence at 1. If a 1 is shifted down two the result would be 8.]

[658] *Atari III* at 14.

[659] This is used only as a turn of phrase and not to suggest that the data stream is a musical work or a sound recording. See *Atari III* at 8.

[660] *Atari II* at 17-8.

- 61 becomes 61131
- 61131 becomes 6113100000 (five 0's representing 1ms of silence)

The full data stream would thus consist of these numbers:
6113100000**72131**00000**83131**00000

10NES "SLAVE/KEY" process on cartridge

a. *Select any number between 1 and 9.*
b. *Multiply that number by 10, shift all characters UP by the number of times this step has been carried out, add the numbers 131 at the end. Wait 1ms.*
c. *Repeat process in step c three times.*

The calculation is the same, except that no seed number is provided.

Examples of the full data stream would thus be:
6113100000**72131**00000**83131**00000 (if the selected number was 5)
4113100000**52131**00000**63131**00000 (if the selected number was 3)
0113100000**12131**00000**23131**00000 (if the selected number was 9)

Verification process

Based on this example, the essence of the operation of the 10NES unlock process works like this, after a cartridge is inserted and the data stream is received:
Step 1: Check if all numbers end in 131. If yes, delete 131 and proceed. If no, reject cartridge.
Step 2: Shift remainder DOWN by a value of 1, 2 and 3 respectively for each number stack.
Step 3: Divide result by 10 and dismiss all decimal values.
Step 4: Check if result is between 1 and 9 for each number. If yes to all, cartridge is verified. If no to any, cartridge is rejected.

From the above illustration, it is clear that a correct data stream could be generated without knowing what the seed number is. Therefore, an acceptable key can be generated without replicating that part of the code that sets the seed number. As long as the start number was between 1 and 9, and the calculation process in the key function is the same, the data stream will be within the prescribed parameters of the 10NES program and will, consequently, accept the cartridge.

On the facts of this case, however, Atari's RABBIT program did not achieve verification based on the predictability inherent in this process. Instead, as Nintendo complained, Atari replicated "the entire song from the 10NES program."[661]

Using the above illustration this means: Atari replicated verbatim both the seed number portion (which is entirely superfluous to this process) and the full prescribed range of start numbers (1 to 9). A working system could have been achieved with less. For example, a non-infringing program would have achieved interoperability by using only one value (e.g. 6) as the start number for generating the key data stream. Knowing that 6 is within the current prescribed range, this would always result in a verifiable data stream. Therefore, those portions of code in 10NES that instruct the machine to select a value between 1 and 9 are not essential to interoperability and need not be copied.

However, Atari submits that it was necessary to copy these instructions to ensure that their cartridges would remain compatible *in future* even if Nintendo changed the code. For example, if RABBIT used only the number 6 to initiate the data stream calculation, and Nintendo discovered this, it could change the code of 10NES to reject any cartridge that sent a data stream calculated based on the number 6 only. To prevent this, Atari needs to use the same instructions and create the data stream using the full range of numbers.

While this is not necessary for *current* interoperability, it is necessary for *future* interoperability. The court in *Atari I* and *Atari II* did not consider this argument. It did, however, indirectly make a finding on it. It held that the portions of code (that would ensure future compatibility) are not necessary for current interoperability and are, therefore, indicative of copying and an objectionable, substantial similarity.

[661] *Atari I* at 5. There are some misleading mistakes in terminology here. What is meant by this contention is that Atari reproduced *all of the code* that generates the data stream (or data song). The data song itself is, of course, not contained in the code and could not be copied from the 10NES program.

Nintendo also complained that Atari replicated the silences between the constituent number series in the data stream *in the exact manner*. In the above example, the silence is 1ms. It is necessary that all data streams should have the same period of silence in order to synchronise communication (it is a mechanism dependant on the rate/speed at which calculations can be carried out by the particular hardware). A silence can be represented by either a number of consecutive 0s or 1s. In this case the silences were always encoded by the 10NES as a series of 0s in all cases. This was deliberately done by Nintendo and the 10NES program code made provision for this.[662] Atari contends that that portion of the 10NES program that decides to encode the silence in either 0's or 1's must be replicated into RABBIT to ensure that, in future, if the 10NES program is changed and now expects the silences to consist of 1's at any point,[663] the data stream generated by Atari's program would still be acceptable.

The court in *Atari III* dismissed these contentions because, "nothing [in case law] suggests a right to copy to ensure *future* compatibility"[664] and the court found no reason to extend the law in this respect. Although the merit of this finding is arguably incompatible with the clear purpose of copyright law expounded in *Atari II* and the separation between idea and expression, this point need not be canvassed further. It does not add to the analysis of the decompilation prohibition because it deals strictly with the scope of fair use based on the exclusion of ideas, methods, procedures etc. It has nothing to do with fair use by decompilation.

---

[662] *Atari III* at 18 where it is said that Nintendo contended that the 10NES program *deliberately* included 0s.

[663] The contention is technically more complex. The argument is that 10NES currently only examines the data stream for a short period, although it is continuously being sent and received. While the silences are being generated, the program code continuously instructs the machine to output 0's. Atari argues that all of the instructions related to this is necessary to ensure future compatibility. It relies on the hypothetical situation that the 10NES system might be changed to check the data stream again at a later point after the initial verification and would then expect the silences to consist of, for example, 1's instead of 0's. Therefore, unless RABBIT contained the same instructions that determine these decisions, the Atari cartridges could become incompatible. See *Atari III* at 20-1.

[664] *Atari III* at 21.

However, the reasons for the court's finding must be examined further because it is responsible for, unwittingly, blending the analysis of interoperability (and the commensurate separation of protected expression and unprotected ideas that are necessary for this purpose) into the decompilation exception.

As explained above, decompilation is considered fair use **because it allows access**. Nothing in the exception implies that it is fair to decompile **because it allows interoperability.** The theoretical basis for the exception is access *per se*, for whatever purpose. Interoperability is merely one of many conceivable indications of the fact that access is fair. That means that interoperability is not determinative of (or even relevant to) the question of whether decompilation is fair. Similarly, the decompilation exception does not prescribe, or in any way influence, what a user may do with the work after it has gained access – this role is performed by general copyright law principles.

However, in *Atari III* the court uses the *fact* that decompilation is necessary to achieve interoperability in order to find that reverse engineering (decompilation) justifies interoperability only to a certain extent. The absurdity of this is obscured by a rather strange public policy argument. The court refers to *Sega* and surmises that it held that "the right to copy portions of program code in a security system must be justified under a fair use analysis"[665] that balances "a presumption of unfairness [arising] from the essentially commercial purpose behind reverse engineering [against] the public policy benefits that flow from allowing third parties to produce independent games".[666]

Consequently, the court finds that "while copying to achieve present compatibility can be justified under the rubric of fair use, allowing preemptive copying to ensure future compatibility would destroy this delicate balance".[667] Clearly the court is wrong in using the "rubric of fair use" that justifies decompilation to determine the extent to which public policy permit reproduction for interoperability.[668] The one has nothing to do with

[665] *Atari III* at 22.

[666] At 21.

[667] At 23.

[668] See Miller *Harvard Law Review* 977 at 1019-1020 where the author states that "[t]here is no principled basis" for the court's use of a "relevant market", one of the factors of the fair use rubric, to determine the lawfulness of decompilation. Regarding the point made here, the same author submits,

the other. Instead, it should have simply asked whether public policy would support an extension of the fair use in ideas and methods in order to achieve future compatibility. Unfortunately, it does not recognise this distinction. Instead, it submits that reverse engineering is only fair insofar as it justifies (which it does not) current interoperability because public policy also favours a proliferation of different games consoles. A right to maintain future compatibility would, in the skewed opinion of the court, undo the competitive advantage a manufacturer has over the closed system of console and cartridge for a short period until the security system is reverse engineered.[669]

Thus, the court concludes:

> "A fair use defense which allows copying for present compatibility balances the incentives for both game developers and console manufacturers."[670]

This finding is problematic. Fair use for interoperability has clearly been established by the courts[671] based on the scope of copyright protection in original expressions. Thus, if an expression is necessary to achieve interoperability, it would be fair to reproduce it in a suitable manner (limited to the essential functions and without copying verbatim unless only one expression is possible) because that expression relates to an idea or process, is not original, and/or, is in the public interest that it should be copied.

Conversely, fair use by reverse engineering in the form of decompilation has been established based on the purpose of copyright. It balances the right to access against the right to reproduce by excluding the *act of reproduction* from the ambit of the act where it is necessary to gain access – *not* where it is necessary to achieve interoperability.

---

at 1020, that "by subordinating the interests of those who create software in favor of those who copy software for their own commercial ends, the court inevitably undercut the incentives that Congress intended the Copyright Act to provide authors of programs" and, consequently, there is no basis for contention that the public interest is relevant when determining the purpose for which decompilation is carried out.

[669] *Atari III* at 23.

[670] At 24.

[671] See the discussion in paragraphs 4 2 3 4 to 4 2 3 7 above.

Thus, the public policy that underpins each of the two fair use instances differ substantively. Unfortunately, *Atari III* finds that the public policy, namely, competition, that supports fair use in the case of one (interoperability) is determinative of the scope of the other (decompilation).

This fundamental mistake is clearly why the statutory decompilation exception would eventually be limited to cases where the process is carried out for the purpose of interoperability only. It completely lost its essential underlying characteristic, namely, fair use for the purpose of access to ideas and, instead, limited it to access for a specific purpose.

Lastly, as a side note to the above analysis, it should be pointed out that the court in *Atari III* refers, throughout, to fair use by "reverse engineering" and not to fair use by decompilation. At this early stage, the term decompilation was not yet in common use. However, it might be said that when the court refers to reverse engineering, it did not mean decompilation but, instead, meant to describe the process of mimicking the functioning of a program by observing the code and writing new instructions that deliver the same result (reverse engineering in the traditional sense). On this construction, the commentary on public policy and interoperability may, just about, be acceptable. In other words, it would mean that the court prohibited the creation of new code that secures future interoperability because it saw this action as outside the scope of permissible borrowing of ideas (by a process of reverse engineering in the simple sense described here).

This construction is wrong for three reasons. First, this would mean that the court disallowed the use of another's ideas for a specific purpose. This is completely at odds with the governing public policy of copyright law that, *per se*, excludes ideas. If an expression is deemed unprotected because it is merely an idea, the whole of that idea is unprotected and not only that part of the idea that is absolutely necessary now. The idea does not become, somehow, protectable because of the reason why it is being used by another. Therefore, it cannot be said that a process of reverse engineering (by writing new code) is unjustified simply because it seeks a future result based on a current idea. Second, the court did not interpret reverse engineering in this sense but, instead, meant engineering in the decompilation sense. This is clear from its specific reference to the judgment in *Sega*, where the court dealt with reverse engineering by decompilation.[672] It is further clear from the fact that *Atari I* and *Atari II* consistently refer to reverse engineering in only one sense, namely, decompilation. It would be ridiculous to assume that the court now opted for a different meaning without making this known. Third, if the court consciously meant to express a view on the fairness of reverse engineering in the traditional sense it would have known that its finding is manifestly incorrect because of the failed logic explained in reason one above. It is highly unlikely that the court, nevertheless, would issue such an opinion.

Regardless, it is clear that the court blended interoperability into the decompilation exception either on purpose or, more likely, unwittingly as a result of its struggle to understand the public interest in interoperability. The fact remains that the decompilation exception was doomed from the start to be unduly, and incorrectly, limited to a specific purpose.

4 2 3 9 *Sega Enterprises v Accolade (1992)*[673]

Judgment by the appeal court in this case (*Sega II*) was delivered after the handing down of *Atari II* but before *Atari III*. The preceding judgment of the district court[674] in *Sega I* was delivered between *Atari I* and *Atari II*. The orders of both courts are reported separately.[675] Therefore, although the judgment in *Sega II* is the first case to establish a fair use exception for decompilation, it was not the first matter to deal with reverse engineering. This makes it more appropriate[676] and more convenient to discuss the *Sega* cases after the *Atari-trio* analysis.

---

[672] *Atari III* at 22-3 with referenced to *Sega.* See the discussion in paragraph 4 2 3 9 below.

[673] *Sega Enterprises Ltd v Accolade Inc* (1992) 977 F.2d 1510 amended and reported as *Sega Enterprises Ltd v Accolade Inc* 1993 U.S. App. LEXIS 78 (*Sega II*). In this work, all references are to paragraph numbers as reported in the original judgment (977 F.2d 1510), unless otherwise indicated.

[674] *Sega Enterprises Ltd v Accolade Inc* (1992) 785 F. Supp. 1392 (*Sega I*).

[675] *Sega Enterprises Ltd v Accolade* 1992 U.S. Dist. LEXIS 16132 as amended, granting the preliminary injunction. The amended order added a recall duty within 10 days. *Sega Enterprises Ltd v Accolade Inc* 1992 U.S. App. LEXIS 24121, vacating the preliminary injunction.

[676] *Atari II* was delivered by the US Court of Appeals for the Federal Circuit (the 13th court of appeals seated in Washington) while *Sega II* was delivered by the US Court of Appeals for the Ninth Circuit (California). The reason for this is the subject matter jurisdiction of the Federal Circuit court in relation

The facts in this case are remarkably similar to that of *Atari*, with one notable exception: no evidence of substantial similarity between the works was submitted. The only literal reproduction and derivative use of code by Accolade related to a minute portion of data.[677] This crystallised the dispute, insofar as the copyright contentions were concerned,[678] to two grounds only, both of which are directly relevant to decompilation: (1) intermediate copying (infringing reproduction of the whole of Sega's program by way of disassembly); and/or (2) making a derivative work (creating a program based on the ideas discovered after decompilation).

The dispute in essence related to the security measures of the Sega program designed to restrict compatibility with the Sega Genesis console to cartridges created by, or under license from, Sega. The security measures were not developed by Sega but derived from a "licensed patent"[679] held by a third party.

This measure, referred to as the trademark security system (TMSS),[680] was only required to authenticate cartridges inserted in a later generation of the console called the Genesis III. The small portion of code relating to the TMSS is the only code copied by Accolade into their program. The TMSS restricted compatibility with the Genesis III console by looking for a set of specific instructions contained in the code on a game cartridge. The "initialization"[681] code consisted of four bytes of data representing the letters S E G and A. If these were found in the code, the console would accept the cartridge as valid and execute the game code.[682]

to patent matters, which was at issue in *Atari II* but not in *Sega II*. Both are, however, circuit courts for the purpose of precedence.

[677] A total of 20 to 25 bytes of data out of between 500 000 and 1.5 million bytes in the contentious work. See *Sega II* at 1516.

[678] Both Accolade and Sega also raised issues in law regarding trademark infringement.

[679] *Sega II* at 1515.

[680] At 1515.

[681] At 1515.

[682] At 1515.

Accolade achieved compatibility between their cartridges and the Genesis consoles by a process of reverse engineering. Accolade twice decompiled[683] the Sega program (before and after the addition of the TMSS portion), from authorised copies of the program extracted from lawfully-purchased Sega cartridges, and analysed the results.

After the first decompilation, Accolade wrote a development manual for all Accolade game developers, containing instructions about how to establish compatibility with the Genesis. These instructions, called the interface specifications, did not contain any of Sega's original code or any substantially similar expression(s). Interface specifications are coded instructions that give effect to system requirements and setting in order to achieve interoperability in a specific environment (for example, between two computer programs or between a program and specific hardware or a network).

After the second decompilation, Accolade discovered the additional code relating to the TMSS. Accolade used this code to write a program header (a set of source code instructions) to be used in all subsequent game cartridges produced by Accolade's developers. The program header was added to the development manual and contained a verbatim reproduction of the whole of the TMSS code, representing a fraction of Sega's code.[684]

In addition to providing a security check feature, the original TMSS code also caused a message to appear on the visual display before gameplay begins. Because this "Sega message"[685] contained the SEGA trademark, and Accolade copied the whole of the TMSS including the portion of code that generated this message, Sega initially approached the court with various trademark infringement claims in terms of the Lanham Act[686] and unfair competition, which are not relevant to this work and will not be discussed. Sega later amended its complaint to include a claim of copyright infringement by way of unauthorised reproduction (making intermediate copies of the

[683] At 1514-6.

[684] *Sega II* at 1516.

[685] At 1515. The Sega Message read: "PRODUCED BY OR UNDER LICENSE FROM SEGA ENTERPRISES LTD."

[686] Us Trademark Act of 1946 (Lanham Act) 60 Stat. 427 15 U.S.C. §1051.

whole and/or reproducing the TMSS code) or adaptation (decompiling the object code into source code) of Sega's code.[687]

In *Sega I* the court opines that the act of decompilation created intermediate copies and that such copies are *per se* actionable because it was made without authorisation and fixed.[688] The court relies on only one case to support its opinion,[689] which dealt with intermediate copying of a motion picture. It dismisses the finding of the court in *Uniden*[690] out of hand, mistakenly stating that this case did not deal with intermediate copying. Regarding Accolade's contention that, in order for infringement to occur, the original work must be compared with the *final* derivative work (as opposed to the interim preparatory works) created by the defendant, the court is equally dismissive. It simply states that it has already established that the act of making an intermediate copy is prohibited and, therefore, the contention can go no further.[691] On this basis, it finds that intermediate copying is not sanctioned by the Act and constitutes the making of an infringing copy.

It then considers Accolade's argument that intermediate copying amounts to fair use in terms of section 107.[692] The court's analysis of the fairness factors is perfunctory at best, influenced primarily by its obstinate refusal to entertain any contention that intermediate copying should be treated any different. This allows the court to ignore

---

[687] *Sega I* at 1396.

[688] At 1396.

[689] *Sega I* at 1397 with referenced to *Walt Disney Productions v Filmation Associates* (1986) 628 F.Supp. 871 at 876.

[690] *Uniden* at fn17. See paragraph 4 2 3 6 above.

[691] *Sega I* at 1396.

[692] The test for fair use in section 107 reads:

> "In determining whether the use made of a work in any particular case is a fair use the factors to be considered shall include -
>
> (1) the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes;
>
> (2) the nature of the copyrighted work;
>
> (3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and
>
> (4) the effect of the use upon the potential market for or value of the copyrighted work."

the second factor "entirely."[693] In relation to the other three factors, the court's brief analysis is entirely determined by its opinion that:

> "The copying at issue here was undertaken by Accolade for financial gain and was aimed at the **creation of a competitive product** which will adversely impact the value of the copyrighted work."[694]

This leads the court to observe that precedent dictate intermediate copying is unfair use. Again, the court based its finding on two cases with no precedential value. It borrows a presumption of unfairness, in the case of commercial use, from the *Betamax*[695] case that dealt with time shifting of television programs, and submits that fair use may not be applied in cases where it will affect the "marketability"[696] of the original work, based on the *Harper & Row*[697] case, which dealt with books.

Therefore, the test for fair use fails because "Accolade's game cartridges compete directly with those of [Sega], which has likely lost sales as a result of Accolade's copying."[698] Two errors of fact are present in this statement. In the first place, similar to the facts in *Atari*, Accolade's game cartridges contained only titles (games) that were *unique* to Accolade. Considering that both parties were in the business of creating and publishing games, not security software, it cannot be argued that Accolade's cartridges compete directly with Sega. The opposite is far more likely, considering that the availability of more games that are compatible with the Genesis consoles would benefit Sega.[699] In the second place, the court assumes that Accolades' (fictional) competition has resulted in lost sales for Sega. This too is a fiction. As the court in *Sega II* observe, no evidence was submitted of harm[700] to

[693] *Sega II* at 1522.

[694] *Sega I* at 1398 (emphasis added).

[695] *Sony Corporation of America v Universal City Studios Inc* (1984) 464 U.S. 417.

[696] *Sega I* at 1398.

[697] *Harper & Row Publishers Inc v Nation Enterprises* at 566-7.

[698] *Sega I* at 1398.

[699] The court in *Sega II* confirmed this. See *Sega II* at 1523-4.

[700] *Sega II* at 1522. See also, at 1518, where the court's reasoning suggest that Sega might have suffered no hardship at all.

Sega's cartridge business because there was none. Sega never raised any contention of loss.

In an attempt to bolster its reasoning, the court in *Sega I* further rejects Accolade's fair use contention because, "this argument implies that 'fair use' should be allowed in order to gain access,"[701] which would do "violence to the term 'access'."[702] This finding is based on the court's view that the express exclusion of ideas from the ambit of copyright[703] in the act does not constitute "an independent basis for the appropriation of protectable expression."[704] In other words, the court argued that the exclusion of ideas is not sufficient reason to find that, where the expression that embodies that idea has been emulated, it is fair use in terms of section 107. Consequently, to allow access to a work in order to discover its ideas cannot be permitted because this would automatically render the 'appropriation' of the expression, which embody that idea, a permitted act.

This conclusion is clearly at odds with the fundamental principles of copyright law, prompting the court in *Sega II* to state that it "verges on the frivolous".[705] The fact that the court chose to ignore long-standing precedent on the exclusion of ideas from protection, and the safeguard against 'appropriation' that is maintained by the law where more than one expression of an idea is possible, in order to force a finding against fair use, is clearly the result of its failure to consider the nature of the work (the second fairness factor). It is, therefore, no surprise that the court in *Sega II* had no difficulty in overturning the court's finding.

The court in *Sega I* made two further findings, both of which are equally misguided. First, it contends that if fair use for *intermediate copying* was to be justified, the act would have been amended to this effect. The fact that the legislator chose not to do

701 *Sega I* at 1398.

702 At 1398.

703 Section 102(b): "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work."

704 *Sega I* at 1398.

705 *Sega II* at 1521.

so at the same time it introduced a *reverse engineering* exception for semiconductor chips, is, so the argument goes, indicative of the intention to exclude fair use for intermediate copying in relation to literary works.[706] Again, in a fashion similar to the reckless commingling of fair use for intermediate copying and fair use for reverse engineering (decompilation) in *Atari III*,[707] the court ignores both legal and technical reality in order to support a pre-conceived notion. The court in *Sega II* dismissed this argument because the decisions of the drafters of the SCPA[708] "says nothing about its intent with respect to the lawfulness of disassembly of computer programs under the Copyright Act."[709]

Second, *Sega I* submits that reverse engineering for the purpose of access cannot be fair use because "alternative methods"[710] exist to achieve the same end. It does not seem to be aware of, or care about, the fact that this is not true in almost all cases – a fact that previous courts have repeatedly confirmed. Instead, it relies on *Atari I* and finds that if Atari persisted in microscopic examination of the chip only, it would have gained lawful access to the work without intermediate copying. This despite the fact that the judgment in *Atari I* contained clear evidence to the contrary – Atari could not access the ideas based only on the object code gleaned from the chip. Furthermore, in *Sega II* the court found that, on the evidence before it, the court "committed a clear error."[711]

This comity of errors, which one may also describe as a comedy of errors, is arguably the reason why the judgment in *Sega II* took care to develop the fair use exceptions for reverse engineering and intermediate copying in far greater detail than *Atari II*. In

---

706 *Sega I* at 1398-9.

707 See paragraph 4 2 3 8 above.

708 Semiconductor Chip Protection Act of 1984 17 U.S.C. ch. 9 §901.

709 *Sega II* at 1521.

710 *Sega I* at 1399.

711 *Sega II* at 152-6 where the court states as follows:

> "Even Sega's amici agree that this finding was clear error. […] Chip peeling yields only a physical diagram of the object code embedded in a ROM chip. It does not obviate the need to translate object code into source code."

Sega II, Accolade raised four contentions to justify its actions.[712] The court, correctly,[713] entertained only the fourth, namely that decompilation to gain access to ideas is a fair use in terms of section 107.

The court begins its analysis of fair use by clearly stating the basis upon which it will find in favour of Accolade. It states that this exception relies on two factors: (1) disassembly is the only means of gaining access to the unprotected ideas; and, (2) a legitimate interest in gaining access has been identified.[714] It then proceeds to apply each factor individually.

4 2 3 9 1 Purpose and character of the use

The court correctly identifies the use of the work during decompilation and states that it was an "intermediate one only and thus any commercial 'exploitation' was indirect or derivative."[715] Furthermore, while commercial use is a factor that weighs against fair use, the presence of a commercial purpose is not an absolute bar to fair use. The

[712] The first three contentions were:

1. That intermediate copying will only be an infringing reproduction if the final derivative work contains substantial similarities when compared with the original.
2. That the exclusion of ideas from protection in section 102(b) is, without more, a warrant to reproduce the text that embody that idea.
3. That disassembly is a permitted act in terms of section 117 insofar as it allows the loading of the program into a computer.

This is a restatement of the contentions, drafted to illustrate the reasons why these arguments had no merit. For the original wording see *Sega II* at 1517-8.

[713] The reasons (at 1518-21) why the court rejected these contentions are:

1. Intermediate copying is, based on the plain language of the act, a reproduction in material form. The court does, eventually, rule that intermediate copying is permissible, but not based on this contention.
2. The fact that the law does not protect ideas does not, *per se*, mean that the work may be copied in order to discover the idea. The right to access the ideas by a process that involves reproduction, rests on another part of the copyright law and not the idea/expression separation.
3. The scope of the exception in section 117 is clearly limited, based on CONTU, to initial installation of the program by an average user only. No other form of copying of code into a machine, in the general sense that would encompass decompilation, was contemplated.

[714] *Sega II* at 1520.

[715] At 1522.

commercial nature of the use is merely a factor, and the degree of commercial use must be considered. On the facts of this case, the court finds that the indirect commercial aspect of use is, at most, of minimal significance[716] and that the public benefit (a proliferation of creative works, i.e. more games being made available) resulting from this use far outweigh the impact a commercial gain to Accolade may have on the fairness test.[717]

As to the purpose and character of use by Accolade, the court states that the evidence makes it clear "its direct purpose in copying Sega's code [by decompiling it], and thus its direct use of the copyrighted material, was simply to study the functional requirements for Genesis compatibility".[718] Therefore, insofar as decompilation involves copying code, it was done for a "legitimate, essentially non-exploitative purpose."[719]

4 2 3 9 2 Nature of the copyrighted work

Unlike the court a quo, the court in *Sega II* conducted a thorough analysis of the second fairness factor because it reflects the fact that not all types of work are "entitled to the same level of protection."[720] In fact, the court not only distinguished between types of work but between copyrighted works of all kinds, even if they are protected as the same type of work. This means that the second factor must be afforded due attention and must have an impact on the determination of fairness. To ignore it, or suppress the significant distinguishing features of the work, would render the fairness test unbalanced.

The court finds that the nature of the work indicates that decompilation should be fair use, and provided three reasons for its finding. In the first place, computer programs are published in a form unlike any other literary work. This form, namely, object code, obscures the unprotected elements of the work in such a manner that it is impossible

716 *Sega II* at 1523.

717 At 1523.

718 *Sega II* at 1522.

719 At 1523.

720 At 1524.

to *read* the work without decompilation.[721] In the second place, the court rejects the contention that a *viable* alternative means exists.[722] Even if the laborious process of attempting to read the object code directly could be considered an alternative, it would still involve the recording of translated object code, albeit by hand.[723] Therefore, decompilation is essential to read the work – a direct consequence of the nature of the work itself. In the third place, decompilation is absolute necessary in order to discover the unprotected functional elements of the work. Even a program that was created in a wholly independent manner (during a clean-room programming procedure), requires that decompilation had, at some previous stage, been carried out in order to provide the interface specifications necessary to work in a clean-room environment.[724]

For these reasons the court finds that decompilation is made necessary by the nature of the work and, although it might entail copying, suggests that it should be fair use. To hold otherwise would vest the copyright owner with a monopoly over functional elements, which would violate the intention of the Copyright Act when it excludes such elements from protection.[725]

4 2 3 9 3 The amount and substantiality of the portion used

Because decompilation, in the opinion of the court, involves reproduction of the whole of the object code (but clearly not the source code), this element weighs against a finding of fair use in theory, although it does not preclude a finding of fairness.[726]

---

[721] At 1525.

[722] At 1526.

[723] At 1525.

[724] *Sega II* at 1526. The only difference between a clean room, and the way in which Accolade worked, is that a clean room maintains an absolute separation between the programmers of the derivative work and the decompiled code. Considering that many of Accolade's cartridges were programmed on commission, and it supplied only the interface specifications to these programmers (and not the code), the process could well be described as a clean room, at least insofar as the works made between the first and second decomplications are concerned. Accolade added copies of Sega's code (the TMSS) to the programmers handbook after the second decompilation.

[725] *Sega II* at 1526.

[726] At 1526.

However, the court states that in this case, “where the ultimate (as opposed to direct) use is as limited as it was here, the factor is of very little weight.”[727] It does not need to elaborate on this point because it already established, in the analysis of the first factor, that the evidence in this case showed that Accolade sought to, and did, create an independent work. Therefore, the substantiality factor may be significantly tempered in a case of decompilation if the *final* derivative work does not contain literal copying or substantial similarity. This looks a lot like Accolade’s first contention that infringement requires a comparison of the original and the final derivate works respectively. However, the fact that, in this case, it is a factor in the fairness test does not mean that it may also be used to influence the test for infringement.

But this does not address the fact that Accolade produced the whole of the TMSS portion and that the court in *Atari II* held that it may not, therefore, qualify as fair use. This is addressed by the court in the *amended Sega II* judgment,[728] where it added a footnote[729] to the original judgment. It explains that the “circumstances are clearly different”[730] because in the instant case “there is no showing that there is a multitude of different ways to unlock the Genesis III console,”[731] unlike the Atari matter where it was shown that the NES console could be unlocked by a variety of coded instructions.[732] Therefore, the TMSS code is purely functional, does not qualify for protection,[733] and may be copied as a whole.

In addition, the court states that even if this was not the case, Accolade would not be liable for copyright infringement in relation to the TMSS code because “we note that

[727] At 1526-7.

[728] *Sega Enterprises Ltd v Accolade Inc 1993 U.S. App. LEXIS 78.*

[729] At 71-2.

[730] At 71.

[731] At 71-2.

[732] See *Atari II* at 18. The relevant footnote text is added in context to the original as part of the analysis of the second fairness factor (the nature of the work). However, the second part of the footnote is more relevant to the discussion of the third factor. For this reason, it fits better in this work as part of the discussion on the third factor.

[733] *Sega II* at 1524.

Sega's security code [which consists of four characters] is of such de minimis length that it is probably unprotected under the words and short phrases doctrine."[734]

4 2 3 9 4 Effect of the use on the potential market

The court interprets this factor to mean that it must consider whether the use will result in "diminishing potential sales, interfering with marketability, or usurping the market"[735] of the protected work in the event that it finds in favour of fair use. The court finds that none of these consequences can be attributed to decompilation where it is conducted simply to enter the market for works of the *same type*.[736] Since the games produced by Accolade and Sega are distinct and in no way competing works, the act of decompilation does not influence the market for Sega's games. All it did do was influence the market for Genesis-compatible games in an "indirect fashion"[737] by breaking a monopoly held by Sega. Maintaining such a monopoly, based on copyright law, by finding that decompilation is unfair, would run "counter to the statutory purpose of promoting creative expression and cannot constitute a strong equitable basis for resisting the invocation of the fair use doctrine."[738] Any economic loss Sega may suffer, where a user elects to buy a title produced by Accolade instead of Sega, would, in the opinion of the court, be minor and insufficient to turn the fairness analysis against Accolade in relation to this factor.[739]

Consequently, the court finds that Accolade is not liable for copyright infringement because its actions amount to fair use.[740]

---

[734] *Sega Enterprises Ltd v Accolade Inc 1993 U.S. App. LEXIS 78* at 72.

[735] *Sega II* at 1523.

[736] *Sega II* at 1523.

[737] At 1523.

[738] At 1524.

[739] At 1524.

[740] *Sega II* at 1527-8. For the sake of completeness, it is mentioned that the court also dismissed Sega's trademark contention. It found the mark in this instance to be functional only and "because Sega did not produce sufficient evidence regarding the existence of a feasible alternative to the use of the TMSS initialization code, it did not carry its burden and its claim of nonfunctionality fails." See *Sega II* at 1532.

In light of this analysis, the court concludes by drafting the decompilation exception as follows:

> "Where disassembly is the **only way to gain access** to the ideas and functional elements embodied in a copyrighted computer program and where there is a **legitimate reason for seeking such access**, disassembly is a **fair use** of the copyrighted work, **as a matter of law**."[741]

It is important to note that, as the highlighted passage shows, the decompilation exception relates to fair use for the purpose of **access** to the work. This makes it clear that the fairness does not, as the court in *Atari III* and *Sega I* would have it, rely on the fairness of intermediate copying. Furthermore, the decompilation exception is qualified by the need to gain access for a **legitimate reason**. The court in this case found that interoperability is a legitimate reason but it *did not* suggest that interoperability is the *only legitimate reason* for decompilation.[742] If this was its intention, there is no reason why it would not have limited the exception to such cases only. Clearly, decompilation for other legitimate purposes exist, such as the (admittedly slim) enjoyment of reading the work, testing the tools used for decompilation, honing the interpretation skills of a programmer or illustrating the process for informative or educational purposes.

The court concludes its finding on fair use with a brief exposition about its perceived impact. It acknowledges that this case affords computer programs "a lower degree of protection than more traditional literary works,"[743] which may rankle with a purist view of copyright that militate for an homogenous approach to all literary works.[744] However, the court supports the different level of protection for computer programs precisely because the nature of the work would otherwise maintain a monopoly far greater than permissible by copyright law in relation to other literary works. To argue

[741] *Sega II* at 1527-8 (emphasis added).

[742] See Samuelson *Journal of intellectual Property Law* 55 where the author agrees that "fair use may be available when decompilation or disassembly is done for other legitimate purposes apart from obtaining access to interface information."

[743] *Sega II* at 1526.

[744] See Miller *Harvard Law Review* 981. See contra Cohen *Southern California Law Review* 1107-11 and the discussion at fn241 above.

that computer programs may not be decompiled, because this is different to how other literary works may be used, is "to force the proverbial square peg into a round hole".[745]

This reasoning is significantly more important than it appears at first blush. Firstly, it recognises that computer programs are an odd fit in the category of literary works. Second, it implies, correctly, that the decompilation exception does less harm to the homogeneity of the category than it would do to sustain the unjustifiable monopoly in computer programs. Therefore, the decompilation exception brought computer programs closer to other literary works, rather than further away. Thirdly, it is not extraordinary to afford some works less, or weaker, protection where it is in the public interest. Where a work is largely functional, the court states, it is "neither unfair nor unfortunate"[746] that it is not entitled to the same protection as a creative work of fiction or some other traditional types of literary work. By differentiating between works and affording each only the level of protection that is warranted, the copyright act maintains the balance necessary to advance progress.[747] In this respect, the judgment in *Sega II* is welcomed because it avoided the common pitfalls of the literary-analogy.

Lastly, for the sake of completeness, the legal position on intermediate copying, post *Sega*, must be summarised.

The court in *Sega II* held that the legality of any form of copying is firstly determined by the exclusive right to reproduction. This requires that only two factors shall be considered: (1) whether there has been a reproduction of the protected expression of another; and, (2) whether that reproduction exists in material form. Thereafter, the legality of the reproduction is determined. If the instance of reproduction is explicitly justified by the Act (based on a statutory exception) the enquiry ends. If no such exception applies, the instance of reproduction may yet be justified by case law.[748]

[745] *Sega II* at 1527.

[746] *Sega II* at 1527.

[747] At 1527.

[748] At 1518.

In the case of intermediate copying, the court finds that it is *per se* prohibited by the Act because it satisfies both elements of the test.[749] Furthermore, intermediate copying is exempted by the Act in certain narrow cases, none of which are applicable to decompilation.[750]

This means that intermediate copying is neither always an infringing reproduction nor is it subject to a general fair use exception. Therefore, the court determines that, where an intermediate copy is made, "authority for such copying must be found in one of the statutory provisions to which the rights granted in section 106 [the exclusive acts] are subject."[751] In other words, the intermediate character of the reproduction is irrelevant and its legality must be determined in the same manner as any other instance of direct copying.

This is as far as the court's analysis of intermediate copying goes. The only other reference thereto is during the discussion of the third fairness factor, where the court held that "wholesale" reproduction (intermediate copying during decompilation) does not add anything toward a resolution of the dispute "because computer programs are also unique among copyrighted works in the form in which they are distributed for public use."[752] That means that the legality of intermediate copies made of computer programs will be influenced by the nature of this type of work. In this context, the nature of the work refers to the fact that object code is not legible.

Thereafter, during the review of the third element of the fairness test, the court finds that the *de facto* infringing nature of intermediate copying is "of very little weight"[753] and outweighed by the other elements which favour access by means of copying.

Consequently, that status of intermediate copying has been changed only where it forms part of the steps taken during decompilation. In such cases, the intermediate

749 At 1518.

750 At 1520.

751 At 1519.

752 *Sega II* at 1525.

753 At 1527.

copies are deemed to be fairly made because it is created in pursuit of a fair use exception for decompilation. This means the reproduction is a fair copy by operation of the decompilation exception. However, this does not mean that the opposite is true. Decompilation is not fair or unfair because of the status of the intermediate copies. The fairness of the intermediate copy is a consequence of the fairness in decompilation, not vice versa.

4 2 3 10 *Sony v Connectix (1999)*[754] and *Sony v Connectix (2000)*[755]

This matter was the last to contribute to the development of the decompilation exception, based on fair use in copyright law, before the emphasis in US jurisprudence regarding decompilation shifted to statutory interpretation of anti-circumvention measures.

Similar to the *Atari* and *Sega* cases, this matter dealt with decompilation of protected code in order to achieve interoperability in relation to video games. However, in this case the interoperability was not between a protected console and allegedly infringing cartridges. Instead, it related to interoperability between protected games and an allegedly infringing alternative to a console. Furthermore, the dispute in this case did not relate to program code that operated as a security measure, but to computer code that controlled the hardware of the console and communicated information between the video game code and the console.[756]

The claimant, Sony, produced the PlayStation (PS) console and either directly, or by way of license, developed produced and distributed PS games that are compatible

---

[754] *Sony Computer Entertainment Inc v Connectix Corporation* (1999) 48 F. Supp. 2d 1212 (*Sony I*).

[755] *Sony Computer Entertainment Inc v Connectix Corporation* (2000) 203 F.3d 596 (*Sony II*).

[756] Such security measures did exist, and were present on both the PS console and the PS games in object code form. These programs, namely the PlayStation Library code and the Wiz Code, were designed to ensure compatibility between the PS console and only genuine PS game CDs. These programs, or portions of code, were not the target of Connectix's activities because it did not seek to achieve compatibility with the PS console. Therefore, these measures were not at issue in this case, were not part of the code reproduced or decompiled by Connectix and no allegation of copyright in relation thereto was raised by Sony. There is no reason why the court in *Sony I* saw fit to include a description of these measures. See *Sony I* at 1215.

with its console.[757] The games were recorded on compact disk (CD), as opposed to ROM chips in a plastic container, and are therefore not referred to as game cartridges in this work.

The defendant, Connectix, did not produce video games or consoles and conducted business as a software developer.[758] It produced a computer program entitled VGS[759] that would enable PS games to be played on a conventional computer instead of the PS console. Because the PS console is a unique type of computer with specialised features and requirements, the VGS program had to mimic the functions of the PS console in order to allow PS games to be compatible with the VGS program. For this reason, the VGS program is referred to as an "emulator"[760] – a term generally used to describe a class of software that perform the function of hardware, and in some cases other software, in an isolated environment.

The core of the dispute relates to the manner[761] in which Connectix created the VGS program and the contention that this amounted to copyright infringement of Sony's computer code stored on the PS console.

In order to create the VGS software, Connectix sought to discover the way the PS console operated and how it controlled the hardware components in the console. This information, which would determine the interface specifications that VGS must meet in order to execute PS games in an analogous manner, was gleaned from an analysis of the operating system, called the Sony BIOS,[762] installed on a PS console. Connectix extracted the code from an authorised copy, obtained by purchasing a PS console which contained the BIOS code on a ROM chip.[763]

---

[757] *Sony I* at 1214.
[758] At 1215.
[759] Virtual Game Station.
[760] *Sony II* at 599.
[761] At 603.
[762] *Sony I* at 1214-5.
[763] At 1216.

The analysis of the BIOS code was aided by two distinct methods of reverse engineering. In the first instance, Connectix copied the object code onto a computer and, with the aid of a debugging program, observed the communications between the BIOS and the VGS program.[764] This stage required that the entire BIOS be copied into the temporary storage of Connectix's computers every time that computer was turned on and the debugging program initiated. During this stage, the BIOS code was not changed or read by humans. It remained in object code form and only its results were observed.

In the second instance, Connectix decompiled "discrete portions of the Sony BIOS"[765] into source code in order to "view directly the ideas."[766] The information obtained in this manner was used to amend and correct the operation of the VGS program[767] which would, eventually, contain new and unique BIOS code written by Connectix.[768] The final version of the VGS program, which was also the first version to be made publicly available, did not contain a reproduction of the Sony BIOS code either directly or indirectly.[769] For this reason, Sony did not contend that the VGS program contained any infringing material.[770]

Instead, Sony alleged that infringing reproductions of the BIOS code was made, in the form of intermediate copies, during both phases of the analysis[771] and that this did not constitute fair use as defined by the court in *Sega II*. Connectix admitted to factual copying of the BIOS code[772] but contended, in *Sony I*, that it was not established that Sony held valid copyright in the work because a filtration exercise had not been carried out to identify the protectable expressions of the work.[773] In *Sony II*, Connectix

---

[764] *Sony II* at 600.

[765] *Sony II* at 601.

[766] At 604.

[767] At 604.

[768] This fact was admitted by Sony before the case started. See *Sony I* at 1214.

[769] *Sony I* at 1219.

[770] *Sony I* at fn7.

[771] At 1217.

[772] At 1215-6.

[773] *Sony I* at 1217.

responded to Sony's contention regarding intermediate copying and submitted that it constituted fair use in terms of section 107.[774]

The court in *Sony I* found against Connectix and granted a preliminary injunction. Although it was not raised by Connectix in the first case, the court considered the fair use defence as part of its determination on Sony's likelihood of success at trial. The opinion of the court in *Sony II* dealt with the appeal, by Connectix, against this judgment and found in favour of Connectix on the basis of fair use.

Regarding the scope and application of the fairness test to the facts of this case, the judgments in *Sony I* and *Sony II* are polar opposites. This makes it convenient to discuss both judgments simultaneously.

4 2 3 10 1 The purpose and character of the use

In a determination of this factor, particularly in cases where a commercial use is present on the facts, both courts used the transformation test as an indicator. It asks whether the use of the work resulted in something different that distinguished its purpose, and therefore its potential commercial use, from the original.[775] *Sony I* found that the VGS program was designed to work as a "substitute" or "replacement" for the PlayStation console[776] and that it could, therefore, not be said to be transformative. It is the court's opinion that VGS sought to rival the PlayStation, which made it a direct competitor. Consequently, *Sony I* found that this factor weighs against fair use.

In *Sony II*, the court came to a different conclusion and found the VGS program "modestly transformative"[777] because it exposed existing PS games to a new platform or operating environment that is different to the console, namely a conventional computer.[778] Furthermore, the court emphasises that the VGS program is itself an entirely new product, consisting of original code. It states, "we are therefore at a loss

---

[774] *Sony II* at 602.

[775] *Sony I* at 1219 and 606.

[776] At 1219 and 1220.

[777] *Sony II* at 606.

[778] *Sony I* at 606.

to see how Connectix's drafting […] could not be transformative, despite the similarities in function and screen output."[779]

Regarding the commercial aspect, the court disagrees with *Sony I* and states that the use of the BIOS code was only intermediate and indirect or derivative.[780] This use was also for a legitimate purpose, namely compatibility, rather than purely commercial.[781] The VGS program does not "supplant" the PS console but creates an alternative, which is also new.[782] Furthermore, the court points out that *Sony I* applied "an erroneous legal standard"[783] in determining the first and fourth factors based on a presumption of unfairness where a commercial purpose has been identified.[784] This presumption had already been rejected by the Supreme Court,[785] and commercial use was reduced to nothing more than a factor that "tends to weigh against fair use."[786] For these reasons, the first factor weighs in favour of fair use.

4 2 3 10 2 Nature of the copyrighted work

Although the court in *Sega* made it clear that evaluation of this factor requires that the unique functional characteristics of the computer programs should be considered, the court in *Sony I* avoided this. Instead, it concentrates on the allegation that, although the BIOS is entirely functional, it is nevertheless a creative work deserving of wide

[779] At 606-7.

[780] At 607.

[781] With reference to *Sega II*, where the court identified compatibility or interoperability as a legitimate use that outweighs the infringement inherent in intermediate copying.

[782] *Sony I* at 607.

[783] At 606.

[784] At 606.

[785] *Campbell v Acuff-Rose Music Inc* (1994) 510 U.S. 569 at 594 where the court held that "no such evidentiary presumption is available to address either the first factor, the character and purpose of the use, or the fourth, market harm, in determining whether a transformative use, such as parody, is a fair one." The same court, at 584, submit that the word "including" in section 107(f), implies that more factors that are not listed in the act may be considered and that no special treatment may be attributed to the two factors that are listed, namely commercial and educational use. In this respect, the court explicitly overruled its earlier finding in *Sony Corporation of America v Universal City Studios Inc*. See the discussion of this case further below in this paragraph.

[786] *Sony I* at 606.

protection. The creativity in this case, according to Sony, is expressed in the structure of the code[787] and was copied when the whole of the BIOS was reproduced during decompilation.

In addition, the court in *Sony I* refused to recognise this as a form of fair use because, in addition to copying the BIOS, Connectix also "**actually used** that code in the development"[788] of the VGS program. This alleged use, in the opinion of the court, goes beyond the scope of fair use set by *Sega* where it was limited to copying for the purpose of studying only.[789] The court in *Sony I* does not provide any argument to support a distinction between use during study and use during development where there was clearly no copying of code beyond the initial decompilation. It seems to suggest that, because Connectix used the BIOS code to determine how the VGS program shall work, this is not fair use. It is difficult to see how it may be fair to study the code but not fair to use the knowledge one has gained but, nevertheless, that seems to be the line drawn by the court. Consequently, it finds that this factor weighs against Connectix.

In *Sony II* the court conducts a more thorough evaluation of this factor and finds that the nature of the work weighs in favour of fair use under the present circumstances. It has no difficulty in dismissing the above argument for a distinction between studying and some other form of "use". It finds that argument to be "unpersuasive" and "unsupported" by the *Sega* judgment[790] because the nature of the work makes it essential to copy the code in order to gain access. Where that need is satisfied by repeated intermediate copying or loading of the code into a program in order to experiment on it, as was the case in *Sega*, it is nevertheless part of the act of studying the code. The court refers specifically to the iterative nature of reverse engineering and, therefore, determine "the semantic distinction between 'studying' and 'use' to be

[787] *Sony I* at 1220.

[788] At 1220 (emphasis added).

[789] *Sony I* at 1220.

[790] *Sony II* at 604.

artificial."[791] Consequently, it refuses to adopt such a distinction for purposes of determining fair use.[792]

This finding makes it very clear that the fair use test for decompilation is not concerned with interoperability only and, where the test is applied, the meaning of interoperability justifies all manner of use of the code short of direct copying of the protected expression. A position made clear in *Sega* and endorsed by the court in this case. The fairness, at least in relation to the second factor, relies on the need to gain access. Thereafter, the "use" of the code is only limited by the justified purpose for which it was accessed. Interoperability is such a purpose, but, unlike the attempt of the court in *Sony I*, the purpose does not become unjustified if, somehow, interoperability is achieved by an examination of the code that goes beyond an initial reading thereof. The line is, as it has always been, whether more than the ideas made its way into the new program.

Furthermore, in light of the fact that fair use for decompilation was "expressly sanctioned,"[793] the court makes is clear that it shall not matter how the disassembly was *technically* carried out. In other words, decompilation *en bloc* or disassembly in a piecemeal fashion are to be treated the same. The court makes it clear that Sega sanctioned decompilation because it was a *necessary* method. It is not possible to argue that the manner in which decompilation was carried out involved *unnecessary* copying and, therefore, render the use unfair. The court clearly states:

> "The 'necessity' we addressed in Sega was the necessity of the method, i.e., disassembly, not the necessity of the number of times that method was applied."[794]

It supports this finding by returning to the nature of the work and its impact on the fairness test. It repeats that the nature of the work necessitates access in order to maintain the public/private interest balance. For this reason, it would be unfair to the user, and tip the balance against public policy, to prescribe a specific method of

---

[791] At 604.

[792] At 604.

[793] *Sony II* at 604.

[794] At 605.

decompilation or impose a restriction on the method of decompilation based on the number of copies required to gain access.[795] To do so would impose a barrier to access that the fairness test is designed to overcome. In this case, for example, it would mean that Connectix had to use the least efficient solution to the problem of access merely because it involves fewer instances of intermediate copying.[796] Therefore, the court has no difficulty in finding that this factor weighs in favour of fair use.

4 2 3 10 3 Amount and substantiality of the portion used

In relation to this factor, the court in *Sony I* again attempted to introduce usage as a consideration. It argues that Connectix "used the BIOS daily" and, even though it did not copy the instructions into the VGS program, it reproduced the essence of the BIOS. Therefore, the factor weighs against fair use because "taking the heart of the original, and making it the heart of a new, is to purloin a substantial portion of the essence of the original."[797]

This is clearly a misapplication of the third factor and the court in *Sony II* does not entertain any of it. It simply reiterates, almost verbatim, the factor as it was outlined and applied in *Sega II* and finds that, although decompilation involves reproduction of the whole of the work, the intermediate nature of such copying means that the factor is afforded "very little weight".[798]

4 2 3 10 4 The effect of the use on the potential market

The finding of both courts in relation to this factor is primarily influenced by its findings in relation to the first factor. In *Sony I*, the court held that the VGS program was a "substitute"[799] for the PlayStation console which will make users less likely to buy the

[795] At 605.

[796] At 605.

[797] *Sony I* at 1221 with reference to *Campbell v Acuff-Rose Music Inc*.

[798] *Sony II* at 606.

[799] *Sony I* at 1221.

console. Therefore, it will have a detrimental effect on the market for the PlayStation which outweighs fair use in this case.

In *Sony II*, the court agrees that there is a likelihood of an adverse impact on Sony PlayStation sales[800] but, because the VGS program was found to be transformative and, therefore, a legitimate market contender, “some economic loss by Sony as a result of this competition does not compel a finding of no fair use.”[801]

The court acknowledges that Sony seeks to control access to PS games through a PS console only, and that the VGS program encroaches on that attempt to control the entire market for PS games. However, it finds that copyright law may not be used or applied to “confer such a monopoly”.[802] Consequently, the fourth factor weighs in favour of fair use despite the potential impact on the market for PS consoles.

The court in *Sony II* concludes that three of the four factors weigh in favour of fair use while one is marginally against. Therefore, the acts of intermediate copying carried out by Connectix during disassembly or decompilation of the BIOS code are protected as fair use in terms of section 107[803] and the preliminary injunction is lifted.

To a large extent, the finding of the court in this case is an endorsement, by repetition, of the fairness test in *Sega*. However, the *Sony* decision also crystalized some details and extended the application of fair use to some degree.

First, it again made it clear that the fairness of decompilation relies on the nature of the work and its inherent obstruction to access. The *de facto* infringement of intermediate copies is, therefore, rendered fair because it serves the fair purpose of gaining access in pursuit of a legitimate purpose. This is a factual finding as much as a legal one. The court in *Sony II* made it clear that intermediate copying *per se* can never reverse the outcome of the test as long as intermediate copying was generally

[800] *Sony II* at 607.

[801] At 607.

[802] *Sony II* at 607.

[803] At 608.

necessary. Therefore, the legality of intermediate copying flows from the sanctioning of decompilation, not the other way around.

Second, the court extended the test by refusing to restrict the manner of decompilation according to necessity. The court made it clear that necessity is the impetus for granting a fair use exception, but, beyond this, the amount of intermediate copying is not restricted to what is absolutely necessary.[804] In fact, there is no further question about what is necessary or not – the act of decompilation is what is sanctioned, not the volume or manner of copying.[805] Therefore, the user is free to deploy any method of decompilation regardless of its efficiency or duplication requirements.

Third, the court introduced the transformation standard as part of the fairness test and applied it directly in favour of close competition, because this serves the public interest.[806] The narrow distinction between the markets of the VGS emulator and the PlayStation console make it clear that copyright law may not be used, via the decompilation fair use test, to maintain or support a monopoly. In fact, the fair use exception is clearly sanctioned even where it seeks to break a monopoly by incremental diversification of the particular market.

This case has been subject to criticism, particularly for failing to attribute sufficient weight to the potential impact on the market or the commercial nature of the use – factors four and one respectively. This, so the argument goes, means that software proprietors are more likely to seek patent protection for the functionality of computer programs in an attempt to prevent reverse engineering that facilitates the creation of

---

[804] See Karas S "Sony Computer Entertainment Inc v Connectix Corp" 2001 *Berkeley Technology Law Journal* 16 33 43-4 where the author submits that a strict necessity test would have caused procedural problems for the district courts by imposing a duty to consider, in greater detail, expert testimony.

[805] See Lee *Marquette Intellectual Property Law Review* 548-9 where the author points out that "courts do not focus on *how* a defendant reverse engineers a program, but only on the fact that the defendant *must* be able to do so" (original emphasis).

[806] By way of increasing the distribution of knowledge to the public. See the discussion in Prestin D "Where to draw the line between reverse engineering and infringement: Sony Computer Entertainment Inc. v Connectix Corp." 2002 *Minnesota Intellectual Property Review* 3 137 140-1 and 173.

a competing program or emulator.[807] However, as the court in *Sony II* clearly states, it is not the purpose of copyright law to shore up a monopoly or enforce a dominant market position. A finding, to the effect that the commercial impact outweighs fair use under these circumstances, would be to bring copyright protection closer to patent protection. This is not only contrary to the purpose and spirit of copyright law but an overstatement of the commercial impact elements of both factors.

This is why the court devotes attention to the meaning of "necessary" in the *Sega* test, emphasising that the need to access the work remains central to the decompilation exception. Applying the commercial impact factors to effectively prohibit access would be tantamount to patent protection by proxy. For these reasons, the finding in *Sony II* is, justifiably, "within the spirit of precedent, reflects the technical and market realities of software engineering, and comports with the institutional competence of the courts."[808] While this is true, it must be noted that the *Atari*, *Sega* and *Sony* decisions represent a departure from the traditional protectionists approach to copyright as a means to stimulate economic endeavour by vesting property right.

This shift is ascribed to the decision of the Supreme Court in *Sony v Universal,*[809] which took a view on copyright law as "a regulatory regime in which interests of copyright owners were to be balanced against the interests of other commercial participants in the marketplace and of the public at large so as to achieve the larger purposes of copyright law".[810] Therefore, the court in *Universal* elected not to enforce copyright law as a property right where it has the effect of controlling the sale of unprotected work. Instead, where technology advances are not readily accommodated within the confines of the letter of the law, the interpretation and application must start anew from the first principles.[811] In *Universal,* this meant that the balance of interests must be achieved by promoting the creation of new work – a sentiment echoed by all

[807] Karas *Berkeley Technology Law Journal* 41.

[808] Karas *Berkeley Technology Law Journal* 44.

[809] *Sony Corporation of America v Universal City Studios Inc* (1984) 464 U.S. 417 (*Universal*).

[810] Samuelson *Journal of Intellectual Property Law* 52-3.

[811] 53.

the decisions above. This is why the courts agreed that the creative incentive, by rewarding the labours of the creator, is not the first principle of copyright law. It stated:

> "The immediate effect of our copyright law is to secure a fair return for an 'author's' creative labor. But the ultimate aim is, by this incentive, to stimulate artistic creativity for the general public good."[812]

In this light, it is clear to see that *Universal* made the development of the decompilation exception possible by emphasising the need to access the work for any further creative process, even if it is at the potential expense of the copyright owner. Under a strictly property-based approach this would not have been possible. As such, the Supreme Court in *Universal* adjusted, substantially, the hierarchical construct regarding the economic and moral justifications for copyright law.

It is understood that the *Universal* decision is subject to critique for its reliance on patent law to draw a line between the purpose of copyright as a regulatory regime and patent protection as an absolute property right. Furthermore, it was not a decision regarding copyright in computer programs and dealt with contributory infringement. However, the Supreme Court in *Vault v Quaid*[813] made the *Universal* decision applicable to cases of decompilation.[814]

---

[812] *Universal* at 432.

[813] *Vault Corporation v Quaid Software Ltd* (1988) 847 F.2d 255 (Vault) at 262. This decision is widely recognised as an anomalous application of the section 117 exceptions. See McManis C R "Intellectual Property Protection and Reverse Engineering of Computer Programs in the United States and the European Community" 1993 *High Technology Law Journal* 8 (1) 25 83-7 for a discussion of this case.

[814] The court in *Vault*, at 269, held that a contractual prohibition on decompilation, contained in a software user license agreement, is unenforceable under state law because it is contrary to section 117(a)(1) which permits the reproduction or adaptation of a program where it is an essential step in using the program. At the time, it was heralded as an equitable result. See Hale D J "Recent Development: Vault Corp. V. Quaid Software Ltd.: Limits To Copyright Protection For Computer Programs" 1989 *Tulane Law Review* 64 270 279. This defence to decompilation was expressly ruled out by the court in *Sega I* (at 1399) and *Sega II* (at 1520) because it goes beyond the intention of section 117 as expressed in CONTU. Nevertheless, both courts accepted the application of *Universal*, as applied by *Vault*, to cases of decompilation insofar as it supports a permissive reading of the fair use test in the interest of public access to work.

4 2 3 11 Other cases after *Sony*

The judgment in *Sony* marked the end of the development of the decompilation exception. Since then, it is said, the focus has shifted to patent enforcement,[815] as predicted by commentators after the *Sony* decision. However, this ignores a number of subsequent cases which have dealt with decompilation to some degree.[816] Although none of these judgments contributed to the fair use exception for decompilation, some decisions, most notably those in the matter of *Oracle v Google*, had some impact on the development of copyright law in relation to decompilation.

The *Oracle* matter did not alter the fair use exception for decompilation, nor did it make any change to how the *Sega* judgment is understood. However, one argument, among the voluminous and contradictory contentions canvassed in this matter, related to the interoperability element in the *Sega* judgment and has led some commentators astray.

The nearly decade-long matter concerns copyright in coded commands that carry out specific individual tasks, such as displaying a document on screen or calculating a

[815] Menell P S "Rise Of The API Copyright Dead?: An Updated Epitaph For Copyright Protection Of Network And Functional Features Of Computer Software" (2018) 31 *Harvard Journal of Law and Technology* 305 310.

[816] See for example: *North American Clearing Inc v Brokerage Computer Systems* (2009) 666 F. Supp. 2d 1299 (dealing with personal liability of a company officer for decompilation in contravention of a license agreement); *Veracode Inc v Appthority Inc* 2013 U.S. Dist. LEXIS 146059 (dealing with the classification of intermediate files as neither source code nor object code in relation to the patent in a decompilation tool); *Versata Software Inc v Infosys Technologies Ltd* 2013 U.S. Dist. LEXIS 186540 and *Dreamcatcher Software Development LLC v Pop Warner Little Scholars Inc* (2004) 298 F. Supp. 2d 276 (both cases dealing with decompilation in contravention of a reverse engineering prohibition in contract); *Texas Instruments Inc v Hyundai Electronics Industries Company* (1999) 190 F.R.D. 413 (drawing a distinction between decompilation and disassembly on technical grounds); *Syntek Semiconductor Company v Microchip Technology Inc* (2002) 307 F.3d 775 (holding that decompiled source code is not necessarily identical to the source code that was compiled, for the purpose of copyright registration); *Oracle America Inc v Google Inc* (finding that the decisions in *Sega* and *Sony* did not create a general interoperability exception to copy necessary code and that the fair use exception does not determine the scope of protection for functional elements); *SAS Institute Inc v World Programming Ltd* (2017) 874 F.3d 370 (that the meaning of the word 'decompile' shall include all forms of analysing the program to learn its details and shall not be restricted to cases where the work has been accessed and recreated).

mathematical angle.[817] Each task was performed by an individual program, written in the Java programming language, and made available to the public in order to stimulate the creation of programs that would be compatible with a software environment based on the Java language. The Java language was conceived by Sun Microsystems and the Java programs at issue were created by Sun Microsystems. A set of programs of this type establish and maintain the communication standard or protocol for interoperability and are, therefore, referred to as application programming interface (API) packages. In this case, the contentious API consisted of two types of code, referred to as the declaring code and the implementing code. The difference is explained as follows, where the word "method" is used to refer to a *program* that is written for the Java platform:

> "In general, to create a new Java method, a programmer must write code that tells the computer both (i) what the method is, including its name, the circumstances in which it should be available to programmers, what types of input data it should accept, what types of output data it should produce, and what types of errors it can generate; and (ii) how to perform the method, including steps for using the specified input data to produce the specified type of output data. The parties refer to the first type of code as "declaring code" and to the second as "implementing code."[818]

At issue in this case was the reproduction of portions of the declaring code, the *names,* of 37 Java APIs[819] and, by doing so, the indirect replication of the sequence, structure and organisation of the Java library file associated with the 37 API packages. Google did not copy the implementing code for any of the 37 APIs, but sought to achieve compatibility between Java programs and its Android operating system by creating its own Java-based platform (the Android API) that could execute programs written by others who used original Java API. In other words, Google's Android operating system incorporated the declaring code only (names of 37 Java API programs) so that an app that was developed using these programs would be compatible with Android because

[817] *Oracle America Inc v Google Inc* (2012) 872 F. Supp. 2d 974 (*Oracle I*) at 977.

[818] *Amicus Brief: Google LLC v Oracle America Inc* No 18-956 2019 U.S. S. CT. BRIEFS LEXIS 4933 at 5.

[819] The detail of all 37 API packages is recorded in appendix D to Menell *Harvard Journal of Law and Technology* 486 et seq.

it would recognise the common declaring code and would rely on the same structural arrangement of these instructions.[820]

Oracle America, the copyright owner of the Java API, first filed suit against Google in 2010 for, among other things, copyright infringement in 11 500 lines of Java API code found to be present in the Android program.[821] The complexity of the matter, which also invoked patent infringement, required the court to split the dispute into three parts, dealing with copyright and patent law issues individually and the matter of damages separately from both.[822] Only the copyright decisions are relevant to this discussion.

In 2012 the matter was heard by a jury trial[823] which found that Google did infringe copyright in relation to 7000 lines of the code copied from the 37 APIs but could not reach a verdict on whether or not Google's conduct constitutes fair use.[824] The jury verdict was followed by an order of the district court which held that the contentious portions of the Java API were not subject to copyright. The court considered the fact that 97% of the API code in Android was not reproduced from the Java API and that the remaining 3% is not suitable for copyright protection because it is dictated by the rules of the Java language and relates exclusively to functions that could only be expressed in one way.[825] The court relies on both *Sony* and *Sega* as support, insofar as these decisions made it clear that interface specifications that are *necessary* for interoperability may be freely reproduced because it is excluded from protection by

---

[820] *Oracle I* at 978.

[821] Two other instances of alleged infringement occurred, but were not pursued beyond the first jury trial. The first related to nine lines of code for the RangeCheck program, incorporated in an early version of Android by a developer who left Sun Microsystems to work at Google. The second related to eight Java files which were decompiled by Google and used to test the Android program during development. None of the decompiled code made its way into the Android program. See *Oracle I* at 983. See also *Oracle America Inc v Google Inc* 2012 U.S. Dist. LEXIS 66417 where the court held that a reasonable jury would consider the wholesale copying of code during decompilation as substantial. It made no finding about fair use.

[822] *Oracle I* at 977.

[823] The details of the parties' submissions to the jury, and Judge Alsopp's instructions, are recorded by Prof Menell. See Menell *Harvard Journal of Law and Technology* 379-385.

[824] *Oracle I* at 976.

[825] At 979 and 982.

section 102(b) as a method of operation or process.[826] The finding is expressly limited to the facts and did not find that APIs are per se unprotectable or that structure and sequence will never be capable of protection.[827]

In 2014 the appeals court[828] overturned this decision on the issue of copyrightability. The court held that the non-literal elements, the structure, sequence and organisations, of the Java API did qualify for copyright protection.[829] The reasoning of the court in this respect is irrelevant to this work, because it dealt only with copyrightability of computer programs and has nothing to do with decompilation.

However, the judgment is instructive insofar as it dealt with Google's contention that the Java API is not copyrightable because it is the means by which Oracle maintains *interoperability* with its Java platform. In *Oracle I* the court held that Google's reproduction of the code and the structure were necessary to achieve interoperability and that only those portions that were essential to this purpose were copied.[830] Therefore, it was influenced by the "compatibility reasoning" in *Sega* and *Sony* when it decided on copyrightability. The appeals court points out "the district court's reliance on *Sega* and *Sony* in the copyrightability context is misplaced"[831] because both cases dealt with fair use and not with copyrightability.[832]

---

[826] At 1001.

[827] *Oracle I* at 1002, issued as a final judgment in *Oracle America Inc v Google Inc* No. 3:10-cv-3561 (N.D. Cal. June 20, 2012), ECF No. 1211.

[828] *Oracle America Inc v Google Inc* (2014) 750 F.3d 1339 (*Oracle II*).

[829] For the sake of completeness, the reasons for the courts finding are summarised here in short form: the work was creative and original (at 1356); non-literal elements are susceptible to protection (at 1356); the expression has not merged with the idea to create a single industry standard, if properly construed at the time copyright vested (at 1362); the work is not excluded under the short phrases doctrine because sufficient originality is present, even in relation to single instructions (at 1363); neither the merger doctrine nor the scenes a faire doctrine are relevant to a determination on copyrightability and should only be applied during an infringement analysis (at 1364); the structural elements are not excluded as a system of purely functional nature because sufficient creativity has been expressed in the structural arrangement and other means existed to achieve the same result (at 1368).

[830] *Oracle I* at 1000-1.

[831] *Oracle II* at 1369.

[832] At 1369.

In other words, it makes it clear that the decompilation exception does not warrant reproduction for the purpose of interoperability and, by doing so, establish an "interoperability exception to copyrightability".[833] The decompilation exception deals only with fair use in relation to a protected work and permits the wholesale reproduction thereof for any justifiable purpose that requires studying of the ideas. Nothing in the decompilation exception determines which parts of the decompiled code may be copied because it either facilitates interoperability or is merely unprotected ideas. The act of decompilation is *per se* fair because it may, inter alia, facilitate the process of establishing interoperability. But it cannot be extended to the point where it warrants copying of code *post* decompilation merely because it is for the purpose of interoperability.

Moreover, even if it could be argued that interoperability is somehow the basis of a fair use defence, it is nevertheless not part of the test for copyrightability in code. The history of the development of the decompilation exception make it clear that it was never about which parts of code shall be protected, or not, based on its utility or function in relation to compatibility. Only in those cases where the decompiled code made its way into the new work, did interoperability become a factor in determining either copyrightability or infringement.

For this reason, the court in *Oracle II* correctly held:

> "Whether Google's software is 'interoperable' in some sense with any aspect of the Java platform **has no bearing on the threshold question** of whether Oracle's software is copyrightable."[834]

This finding is important to the decompilation exception because it put interoperability in its appropriate, and minor, position where reverse engineering is concerned. Firstly, it confirms that the exception was expressly based on the need for access, to any portion of code, because it is necessary to understand the *ideas*.[835] Nothing in *Sega*

[833] At 1370.

[834] *Oracle II* at 1371 (bracketed portion omitted, emphasis added).

[835] See Lee *Marquette Intellectual Property Law Review* at 549 where the author states that, in relation to the judgment in *Sony* and reverse engineering by decompilation, "the underlying reason for the ultimate finding of fair use – is the policy argument."

or *Sony* suggested that access is warranted because it grants access to unprotected code. Access is warranted in relation to all code, regardless of its protected status or not. Therefore, the judgment in *Sega* cannot be said to have made a finding on the protectability of API's at all.

Secondly, the *Sega* exception grants access to the code by way of decompilation, and renders the intermediate copies fair as a consequence. It does not sanction the reproduction of protected expression beyond that stage for any reason, including interoperability. This is made expressly clear in *Atari* and *Sega*. For this reason, these decisions do not enter the dispute about whether or not an alleged infringing work contains a substantially similar reproduction, nor does it require any abstraction-filtration exercise. Decompilation is *per se* fair. Whatever contention regarding infringement is made in relation to the work that was created after decompilation is not influenced by, or relevant to, the decompilation exception.

The *Oracle* matter did not end here, but nothing further in this case dealt with decompilation at all or with interoperability in a relevant manner and is, therefore, not relevant to this work.

For the sake of completeness, the course of the copyright dispute that followed is outlined here and selected notable findings highlighted. The appeal court remanded Google's fair use contention, based on transformative use, for trial, which led to a second jury trial in the district court.[836] At the same time Google's first petition to the Supreme Court following the first appeal was denied. The second jury trial found in favour of Google on fair use, which lead to the second appeal judgment,[837] brought by Oracle, in 2018. Prior to this point, Google had abandoned all arguments based on interoperability.[838]

[836] *Oracle America Inc v Google Inc* No. C10-0356 WHA (26 May 2016) (*Oracle Jury II*). The proceedings before the jury are recorded in detail in Menell *Harvard Journal of Law and Technology* 391-410.

[837] *Oracle America Inc v Google LLC* (2018) 886 F.3d 1179 (*Oracle Appeal II*).

[838] At 1207.

As the court noted in the second appeal:

> "This change in course is not surprising given the unrebutted evidence that Google specifically designed Android to be *incompatible* with the Java platform and not allow for interoperability with Java programs".[839]

The second appeal judgment overturned the second jury verdict and found that Google's conduct does not amount to fair use because, instead of writing its own code to perform the same tasks as the Java API, it copied Oracle's code into the Android program.[840] The court makes it clear that this judgment does not place the defence of fair use beyond the reach of a programmer who seeks to copy code from another in all cases. It recognises that such cases may exist and have been recognised before, notably in *Sega* and *Sony*. However, on the facts of this case, the court finds that the instance of copying is not protected as fair use.[841]

After this case, Google petitioned[842] the Supreme Court again for judicial review of the second appeal judgment and re-introduced the interoperability argument to some

[839] *Oracle Appeal II* at fn11 (original emphasis).

[840] The court states: "Although Google could have furthered copyright's goals of promoting creative expression and innovation by developing its own APIs, or by licensing Oracle's APIs for use in developing a new platform, it chose to copy Oracle's creative efforts instead. There is nothing fair about taking a copyrighted work verbatim and using it for the same purpose and function as the original in a competing platform." *Oracle Appeal II* at 1210.

[841] *Oracle Appeal II* at 1210.

[842] *Google LLC v Oracle America Inc - On Petition For A Writ Of Certiorari To The United States Court Of Appeals For The Federal Circuit* Unfiled (available at https://www.supremecourt.gov/DocketPDF/18/18-956/81532/20190124110509177_Google%20cert%20petition.pdf)

extent.[843] In 2019 the Supreme Court invited the solicitor general[844] to provide a brief to the court on whether or not the petition should be granted. The amicus brief was published on 27 September 2019 and suggested that the petition be denied. The conclusion of the amicus brief puts the dispute in clear perspective regarding copyright law principles:

> "The court of appeals simply endorsed the unremarkable proposition that wholesale copying of thousands of lines of copyrighted code into a competing commercial product for the purpose of attracting developers familiar with the copyright owner's work, while causing actual commercial harm to the copyright owner, is not fair use."[845]

Regarding Google's interoperability argument, the solicitor general finds that it is inconsistent with the meaning of the word "interoperability" in section 1201(f)(4), which is limited to an exchange of information only and does not extend to compatibility between programming techniques (the Java API) and an incompatible new environment (the Android operating system).[846] In other words:

> "More broadly, the fair-use doctrine does not permit copying valuable parts of a work to attract fans to a competing commercial product. Copying 'to get attention or to avoid the drudgery in working up something fresh' disserves copyright's goals."[847]

---

[843] In Google's application it submits that the first appeal court erred when it dismissed the interoperability argument, and contends:

> "Such interoperability was critical for developers programming in the Java language. At Sun's and Oracle's encouragement, developers had invested in learning the Java language and had grown accustomed to using the well-known shorthand commands derived from the Java API declarations. The district court likened those declarations to the keys on a QWERTY keyboard. Developers therefore wanted to use the Java API declarations to write code for Android applications in the Java language. To allow such code to run on Android, Google had to incorporate the applicable Java API declarations. By allowing applications written in the Java language to operate in the new environment, those declarations took on a 'further purpose or different character' in Android that they did not have in Java SE." *Google LLC v Oracle America Inc - On Petition For A Writ Of Certiorari To The United States Court Of Appeals For The Federal Circuit* at 26 (original citation omitted).

[844] *Google LLC v Oracle America Inc* 2019 U.S. LEXIS 3060.

[845] *Amicus Brief: Google LLC v Oracle America Inc* at 22.

[846] *Amicus Brief: Google LLC v Oracle America Inc* at 20.

[847] At 20 (original citation omitted).

The opinion of the solicitor general has not been welcomed by the programming industry and to date 15 amicus briefs have been filed in support of Google's petition, including Microsoft, Mozilla, the Electronic Frontier Foundation, the Developers Alliance[848] and some of the most notable intellectual property scholars Proffs David Nimmer, Peter Menell[849] and Pamela Samuelson,[850] along with 65 other academics.

The *Oracle* matter is not likely to end at this point. However, the "unusual jurisdictional posture"[851] of this case, namely, the lack of legislative interpretation brought about by jury verdicts, the strictly facts-based findings of both appeal judgments and the "limited precedential value"[852] of the case renders any further analysis thereof moot. It has been clearly established that the decompilation exception is not at issue in cases of actual code copying and cannot be abused as a vehicle for a general interoperability exception. If such an exception was to be developed in this case, it would have an independent basis in terms of fair use under section 102.

4 2 4 Statutory amendments for decompilation

In the preceding analysis of case law, it has been shown that the basis for decompilation under American law is the judicially-created doctrine of fair use as applied by the courts on an ad hoc basis.

---

[848] These briefs are indexed and made available online. See Copyright Alliance "Oracle v Google" https://copyrightalliance.org/copyright-law/copyright-cases/oracle-america-v-google/ (accessed November 2019).

[849] Menell P S and Nimmer D "Google LLC v Oracle America: Brief of Professors Peter S. Menell and David Nimmer As Amici Curiae In Support Of Petitioner" https://www.supremecourt.gov/DocketPDF/18/18-956/89422/20190225113504834_37659%20pdf%20Menell.pdf (accessed November 2019).

[850] Samuelson P and Crump C "Google LLC v Oracle America Inc: Brief of 65 Intellectual Property Scholars as Amici Curiae in Support of Petitioner" https://www.supremecourt.gov/DocketPDF/18/18-956/89474/20190225131314910_IP%20Scholars%20Amicus%20Brief.pdf (accessed November 2019).

[851] Menell *Harvard Journal of Law and Technology* 415.

[852] The solicitor general in *Amicus Brief: Google LLC v Oracle America Inc* at 22 point out that the second appeal "does not bind either future Ninth Circuit panels or future Federal Circuit panels in appeals from district courts outside the Ninth Circuit."

However, some statutory amendments have been introduced into the US Copyright Act that are relevant to the protection of computer code in the context of decompilation.

4 2 4 1 The Digital Millennium Copyright Act

In 1998 the Digital Millennium Copyright Act (DMCA)[853] was introduced in Senate as S2037, to amend certain parts of the US Copyright Act[854] and introduced new provisions to implement the WIPO Copyright Treaty[855] and the WIPO Performances and Phonograms Treaty.[856] The primary stated purpose of the DMCA was to give effect to the duty on member states to provide "adequate legal protection and effective legal remedies against the circumvention of effective technological measures that are used by authors in connection with the exercise of their rights."[857]

However, it is clear from the legislative history of the DMCA that its purpose was somewhat less altruistic and sought to establish American anti-piracy law as the guiding force in the development of international and foreign domestic law.[858] In particular, the anti-circumvention measures of the WCT were "modelled after"[859] a report of the Information Infrastructure Task Force established by President Clinton, and the negotiations of the WCT were directly influenced by the chair of the task force.[860] The approach of the DMCA is, therefore, correctly summarised by the US

---

[853] The Digital Millennium Copyright Act Pub. L. No. 105-304, 112 Stat. 2860, 2887 (Oct. 28, 1998) (DMCA).

[854] Sections 108, 112 and 114 and chapters 7 and 8.

[855] WIPO Copyright Treaty TRT/WCT/001 WO033EN 1996 (WCT).

[856] WIPO Performances and Phonograms Treaty TRT/WPPT/001 1996 (WPPT).

[857] Article 11 of the WCT and article 18 of the WPPT.

[858] Negotiations at the time record that Senate was urged, by Bruce Lehman, the Assistant Secretary of Commerce and the Commissioner of Patents and Trademarks, to act because the United States "had a narrow window of opportunity to exercise world leadership by showing our trading partners, through the enactment of potent implementing legislation, that the United States interpreted the treaties to require them to take effective steps to prevent piracy of American property." See Litman J D *Digital Copyright* (2006) 134.

[859] Lee *Marquette Intellectual Property Law Review* 550.

[860] See Litman *Digital Copyright* 122-145 for a comprehensive description of the legislative history of the DMCA and the relation between the WCT and the report of the Information Infrastructure Task Force.

Copyright Office as an attempt "to create a legal foundation to launch the global digital online marketplace for copyrighted works"[861] from the American perspective, in order to "make available via the Internet the movies, music, software, and literary works that are the fruit of American creative genius."[862] The need to fight piracy of American works is central to this ideal.

This led to the addition of chapter 12 of the US Copyright Act by way of the Treaties Implementation Act,[863] which created, inter alia, section 1201[864] to prohibit the circumvention of "a technological measure that effectively controls access to a work protected"[865] by copyright law. This section is intended to supplement[866] the existing rights of copyright owners by imposing restrictions on the manner in which a work in digital form may be used if the owner has applied a technological protection measure (TPM) thereto.

The "strictures" imposed by the DMCA address three distinct areas:

> "Those strictures target not only bad acts (the activity of copying itself), but also bad machines (devices that facilitate copying) and bad services (conduct that enables copying). In this manner, **copyright law expands its reach**."[867]

In other words, the DMCA seeks to address piracy by restricting the essential acts and tools that are necessary to make unauthorised copies of digital works if such works

---

[861] US Copyright Office 2017 *Section 1201 of Title 17: A Report of the Register of Copyrights* i.

[862] Nimmer D "A Riff on Fair Use in the Digital Millenium Copyright Act" 2000 *University of Pennsylvania Law Review* 148 (3) 681 with reference to the senate reports on the DMCA.

[863] WIPO Copyright and Performances and Phonograms Treaties Implementation Act of 1998 Pub. L. No. 105-304 112 Stat. 2861.

[864] The effective date of this section was delayed for two years until 28 October 2000 in order to give the Copyright Office an opportunity to establish additional temporary exceptions and declare exempted types of work. Some commentators also suggest that the delay was necessary to "allow the development of a sufficient record as to how the implementation of these technologies is affecting availability of works in the marketplace for lawful uses." See Nimmer *University of Pennsylvania Law Review* 699.

[865] §1201(a)(1)(A).

[866] US Copyright Office *Report of the Register of Copyrights* i.

[867] Nimmer *University of Pennsylvania Law Review* 684 (emphasis added, footnotes omitted).

are protected by a TPM. This is done by establishing a new restricted act in relation to copyright-protected work, namely, the act of circumvention, ungoverned by copyright law.

To "circumvent a technological measure" is defined widely to include the decryption, descrambling, avoidance, removal, deactivation or impairment of a technological measure without authorisation of the copyright owner.[868]

In *Universal City Studios v Reimerdes*[869] the court held that a computer program falls within the meaning of a technological measure. This judgment was confirmed by the Supreme Court in *Universal City Studios v Corley*.[870] Which means that any action to manipulate a computer program that controls access to another program is prohibited by section 1201 *and* that any computer program designed for such a purpose is a subject to the restrictions on manufacture and distribution of circumvention devices and tools. In other words, section 1201 applies equally to the act of decompiling a program that controls access (or compatibility with another program, such as the 10NES or SEGA program) and the decompilation tools such as a disassembler application.

Section 1201 makes the "act of circumvention" a civil offence for which injunctive relief, actual and statutory damages,[871] impounding of any device or product, costs and/or remedial modification or destruction may be ordered in terms of section 1203.

In order to temper the potentially adverse effects of the anti-circumvention measures in relation to lawful uses of the work, where a party "may have a legitimate need to

[868] §1201(a)(3)(A), see also the additional definition in §1201(b)(2)(A) which contain similar wording.

[869] *Universal City Studios v Reimerdes* (2000) 111 F. Supp. 2d 294 at 317 (*Reimerdes*).

[870] *Universal City Studios v Corley* (2001) 273 F.3d 429 at 434 (*Corley*).

[871] Actual damages may include any profit derived from the violation. Statutory damages are set at between $200 and $2500 per act of circumvention.

engage in circumvention,"[872] a number of balancing provisions are included in section 1201,[873] two of which are relevant to this work.

4 2 4 1 1 DMCA exceptions

First, the prohibition on circumvention does not "affect rights, remedies, limitations, or defences to copyright infringement, including fair use."[874] This is not an exception to the prohibition, nor does it offer a right to engage in circumvention – its only effect is to leave intact the existing defences, including the fair use exception for decompilation.[875] The legislative history of the DMCA makes it clear that it is not intended to amend the fair use doctrine in section 107 in any way.[876] Where the provisions of the DMCA necessitate additional measures to balance user interests, the DMCA introduced new exceptions (one of which is discussed below) which are not governed by section 107. Therefore, this section *cannot* be used to justify circumvention for any purpose, even if that purpose is an established form of fair use, because "the fair use doctrine has never given anyone a right to break other laws for the stated purpose of exercising the fair use privilege".[877]

Second, the DMCA introduced a permanent exception to the prohibition specifically for reverse engineering.[878] This exception permits the circumvention of a computer program, or any other technological measure, that controls access to another program

[872] US Copyright Office *Report of the Register of Copyrights* i.

[873] See Samuelson P "Intellectual Property and The Digital Economy: Why the Anti-Circumvention Regulations need to be revised" 1999 *Berkeley Technology Law Journal* 14 519 537-543 for a discussion of the development of the DMCA exceptions.

[874] Section 1201(c).

[875] See Nimmer *University of Pennsylvania Law Review* 716 where the author points out, with reference to the House Judiciary Committee's report on the DMCA:

> "The upshot is that fair use would apply only following lawful access, not as a basis for obtaining such access in the first instance. An individual would not be able to circumvent in order to gain unauthorized access to a work, but would be able to do so in order to make fair use of a work which he or she has acquired lawfully" (original citation and quotation marks omitted).

[876] Nimmer *University of Pennsylvania Law Review* 722-3.

[877] Samuelson *Berkeley Technology Law Journal* 539 with reference to the Judiciary Hearings on the DMCA.

[878] Section 1201(f).

"for the **sole purpose** of identifying and analysing those elements of the program that are necessary to achieve **interoperability**"[879] with other programs.

While it is true that this exception is intended "to ensure that the judicial extension of fair use to reverse engineering not be undercut,"[880] the view of the Copyright Office, that "the overall goal of section 1201(f) was to preserve the ability to engage in the activities found to be noninfringing by the Ninth Circuit in the *Sega Enterprises Ltd. v. Accolade, Inc.* decision,"[881] is false. It has been shown above that the court in *Sega* **did not subject** the decompilation exception to an interoperability limitation. Instead, it permitted reverse engineering for any legitimate purpose. Therefore, it is a misrepresentation[882] to suggest that section 1201(f) seeks only to perpetuate the activity sanctioned by *Sega* when, in fact, the DMCA exception is significantly narrower.

Furthermore, section 1201(f) is an exception to the anti-circumvention prohibition, not an exception to copyright infringement.[883] The adverse effect of the prohibition on fair use is, however, not adequately addressed by the exception because it misread the *Sega* decision. This means that circumvention may only be carried out to reverse engineer where it is done for the sole purpose of interoperability, while the decompilation exception warrants reverse engineering for any legitimate purpose.

As a consequence, it is not copyright infringement to reverse engineer for the purpose of, for example, self-improvement, entertainment or inspecting vulnerabilities to

[879] Section 1201(f)(1).

[880] Nimmer *University of Pennsylvania Law Review* 702.

[881] US Copyright Office *Report of the Register of Copyrights* 15.

[882] See the analysis of the *Sega* decisions above, particularly para 4 2 3 9 4 , where it is clear that the court did not impose an interoperability limitation. See *Sega II* at 1527-8, where the court defines the exception as follows: "Where disassembly is the only way to gain access to the ideas and functional elements embodied in a copyrighted computer program and where there is a legitimate reason for seeking such access, disassembly is a fair use of the copyrighted work, as a matter of law."

[883] Burk D L "Anticircumvention Misuse" 2003 *UCLA Law Review* 50 1095 1107.

malware,[884] but this action may not be carried out if the work is protected by a TPM. In other words, it is an anti-circumvention offense to perform an act that is clearly established as fair use. As some commentators observe, this has effectively established a new "right of access,"[885] or an "entitlement to exclude access to ideas,"[886] as part of the copyright owner's exclusive rights because it failed "to anticipate any new or unexpected reason that users might legitimately have for needing access to a work."[887]

---

[884] The DMCA does contain permanent exceptions for encryption research (section 1201(g)), some cases of security testing (section 1201(j) and archiving (1201(d)). See the discussion below and Shemtov *Beyond the Code* 93 and 95 for more examples of legitimate reasons to decompile that are prohibited as a result of the DMCA.

[885] Burk *UCLA Law Review* 1106. This effect is most notable in relation to the anti-competitive behaviour that followed the promulgation of the DMCA and allowed proprietors to "block aftermarket competition in laser printer toner cartridges, garage door openers, videogame console accessories, and computer maintenance services." See Electronic Frontier Foundation 2014 *Unintended Consequences: Sixteen Years under the DMCA* 1-2 and the body of the report detailing the many instances of DMCA abuse.

[886] Shemtov *Beyond the Code* 93 and 95.

[887] Burk *UCLA Law Review* 1106. See also Samuelson *Berkeley Technology Law Journal* 543-6 where the author lists five examples of innocent or legitimate use that do not amount to copyright infringement but are, nevertheless, obstructed by section 1201. See further Samuelson P and Scotchmer S "The Law and Economics of Reverse Engineering" 2001 *Yale Law Journal* 111 1575 1642 where the authors submit:

> "There are also many reasons to reverse-engineer technical protection measures to enable other reasonable follow-on uses of technically protected digital content: analyzing technical measures used to hide infringing copies of copyrighted works, analyzing technical measures used to hide stolen trade secrets or other confidential information, analyzing a virus program wrapped in a technical measure, creating backup copies of software or data, restoring a rightful copy after the crash of one's hard drive, preserving information (e.g., evidence of some illegal activity), preventing surveillance of a licensee's business activities, preventing technical "self-help" measures from being wrongfully invoked, bypassing country codes in a product so one can play a DVD movie for which one has already paid the standard fee on one's DVD player, bypassing controls that prevent users from fast-forwarding through a movie, and making other fair uses, such as excerpting clips from technically protected movies to demonstrate that a particular word (e.g., "redskins") has been used in a derogatory fashion."

In addition, because section 1201(f) is inherently limited to interoperability only, it does nothing to cure the unconstitutionality[888] of the circumvention prohibition and, furthermore, does violence to the impetus for the fair use decompilation exception. The DMCA extends protection to works far beyond the limits imposed by copyright law by, inter alia, restricting the reproduction of work after the expiration of the term of copyright, effectively prohibiting the reproduction of works that are not subject to copyright protection (such as unoriginal compilations) and effectively extending copyright protection to ideas, concepts and other unprotectable elements in a work.[889] Considering that the impetus for the decompilation fair use exception is, and should remain, *access* to the ideas of the work, section 1201 places a significant restriction on the scope of the fair use exception by obstructing access to the entire work unless it is done for the purpose of interoperability.

This creates an internal conflict between two parts of the copyright regime:

> "It is suggested that the current position under our copyright regimes regarding decompilation is self-contradictory: while one part of this regime provides that ideas and concepts are open for the public to study and use, another part effectively provides that where computer programs are at issue, the only effective manner for gaining access to certain ideas and concepts constitutes copyright infringement unless done for the limited purpose of achieving interoperability. Thus, one branch of our copyright regime appears to defeat an explicit objective of another branch of the same regime."[890]

The court in *Corley* briefly considered the constitutionality of the DMCA and dismissed it because "fair use has never been held to be a guarantee of access to copyrighted material in order to copy it by the fair user's preferred technique or in the format of the

---

[888] The overbroad extension of copyright protection, by operation of the anti-circumvention provisions, exceeds the power vested in congress under the copyright clause of the constitution. See Burk *UCLA Law Review* 1107 and 1108 where the author states:

> "The controlled content may include uncopyrightable facts, public domain materials, or purely functional works, yet unauthorized access will constitute just as much of a violation as it would if the content were copyrightable original expression."

[889] See Burk *UCLA Law Review* 1108.

[890] Shemtov *Beyond the Code* 95.

original."[891] This finding is limited to the facts of the case which dealt exclusively with the prohibition on dealing in circumvention tools. For this reason, the court refused to entertain the argument that the DMCA makes most instances of fair use impossible. Therefore, it did not consider whether or not the DMCA is unconstitutional for effectively extending copyright protection beyond its permissible boundaries.

Consequently, the issue remains unanswered and the status quo, for purpose of this work, is unchanged - decompilation under American law is fair use for any legitimate purpose, but, if the work is protected by a TPM, it may only be decompiled for the purpose of interoperability.

4 2 4 1 2 DMCA cases

A review of the cases that dealt with the reverse engineering exception in the DMCA make this point clear. In all of the cases, the court has refused to apply the exception for any purpose other than interoperability.

4 2 4 1 2 1 *Lexmark International v Static Control Components (2004)*[892]

In this case, the appeal court considered whether Static Control Components (SCC) was liable for, inter alia, violating the DMCA prohibition on circumvention[893] and the prohibition on dealing in circumvention devices.[894]

SCC created and sold to the public a device, the SMARTEK chip, which enabled users of the microchip to manufacture printer ink cartridges that would be compatible with Lexmark printers.[895] Lexmark contended that this constituted an act of circumvention of a TPM, namely the Toner Loading Program and the Printer Engine Program,[896] which were responsible for controlling access by checking that only original Lexmark cartridges would be accepted by a Lexmark printer. In addition, Lexmark contended that SCC was guilty of the trafficking violation in the DMCA by trading in the SMARTEK

[891] *Corley* 459.

[892] *Lexmark International Inc v Static Control Components Inc* (2004) 387 F.3d 522 (*Lexmark*).

[893] Section 1201(a)(1)(A).

[894] Section 1201(b), specifically the trafficking prohibition.

[895] *Lexmark* at 530-1.

[896] At 529-530.

chip. The chip contained, inter alia, a verbatim copy of the code of the Toner Loading Program, which was necessary to make the chip compatible with Lexmark printers,[897] obtained directly from a publicly-accessible copy contained in the memory of all Lexmark printers.[898]

The court dismissed the DMCA contentions because it found that the programs did not constitute a TPM in terms of the Act and, therefore, SCC was not liable for any act of circumvention or trafficking in a circumvention device. The Lexmark programs were used to check for compatibility between printers and cartridges, they were not used to prohibit access to a copyrighted work.[899] Therefore, no claim of circumvention may follow because SCC did not circumvent a TPM in the making of the SMARTEK chip. The fact that the SMARTEK chip achieved the result of rendering ineffective the Lexmark compatibility measures (the programs), did not constitute an act of circumventing a TPM, because the Lexmark programs were not TPMs[900] and were not protected works.[901]

Up to this point, the judgment is unremarkable. However, in obiter dicta, the court, by way of one dissenting and two consenting opinions, remarks on the likelihood that the reverse-engineering exception in the DMCA may be applicable to SCC's conduct. These remarks are based on the hypothetical scenario that the Lexmark programs did constituted TPMs.

On this assumption, the court opines that the reverse engineering exception may be available to SCC because it was seeking to create an independent program.[902]

[897] *Lexmark* at 530.

[898] At 546-7.

[899] *Lexmark* at 547. The code of the programs were open and unsecured, stored in the memory of all Lexmark printers.

[900] See *Lexmark* at 548 where the court held: "Because Lexmark's authentication sequence does not restrict access to this literal code, the DMCA does not apply." See also *Lexmark* at 550.

[901] *Lexmark* at 544 where the court found that the programs were not eligible for copyright protection because it did not embody sufficiently original expression and were likely also excluded under the *de minimus* test.

[902] *Lexmark* at 550.

However, the court does not depart from the limitation imposed by the interoperability requirement in the DMCA. It only clarifies the exception insofar as the meaning of interoperability is concerned. Judge Sutton, for the majority, states that, when enacting the exception:

> "Congress added the interoperability provision in part to ensure that the DMCA would not diminish the benefit to consumers of interoperable devices in the consumer electronics environment."[903]

Therefore, the court opines that the DMCA anti-circumvention provisions may not be used to establish a monopoly over compatible products. The interoperability exception is intended to perpetuate the accessibility of work. Where a TPM is circumvented, the DMCA should not be read to impose a burden of proof on the defendant that their conduct is non-infringing. In other words:

> "We should be wary of shifting the burden to a rival manufacturer to demonstrate that its conduct falls under such an exception in cases where there is no indication that it has any intention of pirating a protected work."[904]

This opinion by concurring Judge Merritt is important because it emphasised the role of the savings clause for fair use in section 1201(c). It makes it clear that the intention of the DMCA was to indirectly protect copyrighted work against piracy,[905] not to define, limit or extend the scope of protection of those works.

In this context, the court emphasises that the right to make competing products remains unaffected by the DMCA, unless it is shown that a TPM was circumvented and that the relevant TPM was applied to control access to a protected work. Where, as in this case, neither of these requirements are met, the DMCA may not be abused to render an attempt to enter the market an act of circumvention. Unfortunately, the court stops short of extending this reasoning to the point where it redefines interoperability. All it did was to make it clear that interoperability is a justifiable purpose for circumvention because it accords with the primary goal of the DMCA.

---

[903] *Lexmark* at 549 (inline quotation omitted).

[904] At 552.

[905] At 552.

4 2 4 1 2 2 *Davidson Associates v Jung (2005)*[906]

In this matter, the first case to consider the reverse-engineering exception of the DMCA directly, the court imposed a devastating restriction on fair use for the purpose of interoperability – directly opposite to the sentiments of the court in *Lexmark*, which also severely limits the decompilation exception.

The appellant conducts business as Blizzard Entertainment Inc (Blizzard), creator and owner of copyright in a number of computer games.[907] Blizzard is also the creator of an online gaming platform dedicated to games in the Blizzard stable. The platform, entitled BATTLE.NET, allows the owners of lawful copies of the games to interact and access unique features available only on the platform.[908] The platform also contained a security and compatibility feature,[909] which ensured that only users in possession of a lawful copy of a Blizzard game would be able to access the platform.

Jung and others were responsible for creating, operating and making available the code for, a competing online gaming platform called BNETD.ORG. This platform was intended to emulate the operation of the Blizzard platform and allow users access to the online features of the games without the need to use the Blizzard platform. It was created with the help of, inter alia, decompiled code and analysis of the BATTLE.NET program and several instances of reverse engineering.[910] The end-user licenses and terms of service associated with all Blizzard games and users of the Blizzard platform specifically prohibit decompilation, reverse engineering in general and the making of derivative works.[911] The BNETD.ORG platform mirrored the functions of the Blizzard platform, but it did not contain the same authentication processes. Instead, it contained

[906] *Davidson Associates v Jung* (2005) 422 F.3d 630 (*Davidson*).

[907] The dispute concerned Blizzard games *StarCraft*, *StarCraft: Brood War*, *WarCraft II: Battle.net Edition*, *Diablo*, and *Diablo II: Lord of Destruction*.

[908] *Davidson* at 633.

[909] The feature required the user to enter a unique authentication code (the CD key) attached to each original copy of a Blizzard game. The platform verified the accuracy of the code before permitting access. See *Davidson* at 633-4.

[910] *Davidson* at 636.

[911] See fn 4 and 5 of the judgment for the original wording of these agreements.

a modified version that would recognise a CD key even if it was invalid.[912] This allowed users to access the online features of Blizzard games by using an unauthorised copy of the Blizzard games. The respondents also made the code of the BNETD.ORG platform available to the public, from which further emulators of the Blizzard platform were created by others.

For these reasons, Blizzard claimed copyright infringement and unlawful circumvention of the TPM measures applied to restrict access to the Blizzard online platform.

Regarding copyright infringement, the court finds Jung liable because the acts of reverse engineering were expressly excluded by contract in the terms of the EULA and terms of service.[913] The court distinguishes this case from the judgment in *Vault*, where it was held that a state law that conflicts with the provisions of the US Copyright Act is unenforceable. It finds that parties are free to waive their fair use defences or rights to reverse engineering.[914]

Regarding the circumvention violations, the court states that the exception in section 1201(f) does not excuse the actions of Jung in this case, because it constituted infringement. To understand the mistake in the court's reasoning, it is necessary to reproduce in full the court's summary of this provision:

> "To successfully prove the interoperability defense under § 1201(f), Appellants must show: (1) they lawfully obtained the right to use a copy of a computer program; (2) the information gathered as a result of the reverse engineering was not previously readily available to the person engaging in the circumvention; (3) the sole purpose of the reverse engineering was to identify and analyze those elements of the program that were necessary to achieve interoperability of an independently created computer program with other programs; and (4) the **alleged circumvention did not constitute infringement**."[915]

[912] *Davidson* at 636.

[913] At 639.

[914] *Davidson* at 639. In the UK, this is not possible. The impact of a contractual restriction on decompilation is discussed in more detail in paragraph 5 4 3 below.

[915] *Davidson* at 641-2 (emphasis added).

The first three factors correctly summarise the wording of the section. However, in relation to the fourth factor, the wording of section 1201(f) is significantly different. It states that circumvention is permitted "to the extent any such **acts of identification and analysis** do not constitute infringement **under this title**."

The difference is twofold: First, the exception states that it is the act of identification and analysis which must be free of infringement. The court states that it is the act of circumvention itself that must be non-infringing. Second, the exception determines infringement with reference to "this title", namely, the US Copyright Act. The court omits the last three words, leaving the impression that if the act of circumvention is an infringement of anything, the exception may not apply.

The cumulative effect is that the court applies an incorrect and overly-restrictive test for permissible reverse engineering.[916] This is made clear in its findings on the facts. The court conducts no further analysis after the above summation, and simply states that the exception is not available because Jung's "*circumvention* in this case constitutes infringement"[917] because it allowed unauthorised games to be played in an online environment.[918] It must be noted that this was only possible on the BNETD.ORG platform – the emulated environment. In no way did Jung's actions or the acts of circumvention make it possible to play unauthorised games on Blizzard's platform. Furthermore, nothing in this case concerned the making of unauthorised reproductions of Blizzard's games. Jung and the BNETD.ORG platform played no role in the making of such copies – it merely allowed such work to be used in an online environment over which Blizzard had no control.

[916] See further the case discussion by AH Rajani "Davidson & Associates v. Jung. (Re)interpreting Access Controls" (2006) 21 *Berkeley Technology Law Journal* 365 at 376, where the author submits that the court "blended its analysis of one issue into others" and, as a result, the court's analysis of the second factor in particular delivered a "murky opinion with unclear precedent." See further at 377 where the author states that the court "failed to articulate that when an access control is used in more than one way – for example, during software installation and then again to regulate online access – determining what the access control actually controls should be based on the context of its use."

[917] *Davidson* at 642 (emphasis added).

[918] At 642.

In this context, the reasoning of the court seems alarmingly incorrect. And, to a large extent, it is indeed lamentable. However, the Blizzard games also contained instructions that, when played in an online multiplayer environment, activate Battle.net mode features.[919] These mode features were not otherwise accessible to users of the games.

The BNETD.ORG program emulated the Blizzard platform to the extent that it allowed users of games to access the mode features of the games despite the fact that they were not connected to the Blizzard platform.[920] Therefore, it allowed access to the mode features without the need for a verified CD key, effectively circumventing this access control measure applied by Blizzard.

In light of these facts, it is clear that the BNETD.ORG program does circumvent a TPM. This is not, however, a reason to suggest that the act of reverse engineering to create the program falls outside the scope of the statutory exception.

The BNETD.ORG program was clearly made for the purpose of interoperability, analogous to the facts in *Sony*. Jung circumvented the Blizzard TPMs to identify and analyse the program code and did create an independent program. This meets all three of the exception requirements.

In relation to the fourth, the act determines that the "acts of identification" may not "constitute infringement under this title". Clearly, the act of analysing the code of the Blizzard program does not constitute an infringement of copyright law. This is exactly why the decompilation prohibition was developed – to ensure that such actions would remain lawful.

---

[919] At 633. The mode features are described by the court as follows:

> "Battle.net mode allows users to create and join multi-player games that can be accessed across the Internet, to chat with other potential players, to record wins and losses and save advancements in an individual password-protected game account, and to participate with others in tournament play featuring elimination rounds."

[920] *Davidson* at 640.

However, on the construction of the court, Jung's actions are not exempted because an act of circumvention occurs at a distance, between the user of an unauthorised copy and the mode features recorded on that copy. The fact that the Jung platform allows this to happen, incidentally to its lawful primary purpose of interoperability, is sufficient for the court to find that it is an act of circumvention which also constitutes infringement.

This construction is substantially flawed. Its effect is that, where an act of circumvention has been shown, and that act can be connected to unlawful access to a protected work by any party, the reverse-engineering exception is not available. It also suggests that the fourth element is an internal, DMCA-only test. According to the court, it asks whether the act of circumvention is an infringing act in terms of section 1201 (not in terms of title 17 as a whole) and, if so, the defence must fail. Considering that all acts of circumvention are per se infringing violations of the DMCA provisions, this means that the defence will always fail – rendering it entirely dead letter.[921]

Such circuitous reasoning cannot be the intention of the court. Its superficial examination of the exemption makes it, however, difficult to assume that it considered the implications of its constricted reading of the act. Even if one is to assume, as it should be, that the fourth factor is not an inward-facing test for infringement based only on section 1201, it still does not explain why the court considered the actions of Jung to be an infringement of copyright law and, therefore, an inexcusable act of reverse engineering.

The exemption is clearly intended to limit reverse engineering where the act of studying the work does not constitute a copyright infringement – that is the wording of section 1201(f)(1). This is nothing more than a statutory limitation to prevent the reverse-engineering exception being used to create infringing copies or adaptations.

---

[921] See further Rajani *Berkeley Technology Law Journal* 365 at 380 where the author submits that "the court in effect interpreted interoperability as a feature that could be locked and unlocked by an access control; that is, the court failed to distinguish between the videogame software and services offered for that videogame." Thus, as the author states at 383, "that an act of reverse engineering was prohibited by Blizzard's ELA should not automatically transform BnetD's conduct into circumvention."

However, the court in this case apply this limitation (or a redacted and amended version thereof), to suggest that the reverse-engineering exception may not be used if the program that resulted from it can be used by another to circumvent a TPM. This mistake relies not only on the wrong wording, but is also *ultra vires* section 1201.[922]

The exception deals specifically with the “acts of identifying and analysis” as the factor for determining infringement, not the act of circumvention, whether by the claimant or a third party. The unfortunate consequence is a confusing impression that, even where reverse engineering is conducted for interoperability, the DMCA will obstruct it if the derivative work *might* be used for the purpose of circumvention.

4 2 4 1 2 3 *Sony Entertainment America v Divineo (2006)*[923]

This case was an application for summary adjudication and default judgment regarding violations of section 1201(b) - the prohibition on dealing in circumvention devices referred to as the trafficking offences under the DMCA.

Insofar as the court dealt with the interpretation of section 1201(a)(2) and 1201(b) and the meaning of a device that circumvents an effective protection measure, the judgment is tangential to this work. However, an examination of the opinion of the court is important to this work where it relates to the appropriate application of the reverse-engineering exception as a defence. In this regard, it corrects the faulty impression created by the court in *Davidson*.

The plaintiff Sony is the same party described in *Sony* above.[924] It manufactures and distributes PlayStation (PS) consoles and PlayStation compatible games. At issue in this matter was the authentication process carried out by the PS console when a game disc is inserted. In order to process the content on the disc, the console must detect the presence of a unique code that is only found on original and authorised discs. The

[922] See further Rajani *Berkeley Technology Law Journal* 365 at 378-9.

[923] *Sony Computer Entertainment America Inc v Divineo Inc et al* (2006) 457 F. Supp. 2d 957 (*Divineo*).

[924] See paragraph 4 2 3 10 above.

unique security code is not copied when an original disc is reproduced without authorisation, resulting in the PS console rejecting the copied game disc.[925]

The defendant Divineo, produced and offered for sale on its website, a computer program entitled HDLoader and a number of modification chips (the mod chips) for the PS console. The HDLoader program allows users to make unauthorised copies of PS games onto external hard drive storage devices connected to the console. The mod chips, when installed in a PS console, ensure that the authentication process will not reject a disc if it fails to detect the original security code.[926]

Sony contended that both the HDLoader program and the mod chips are devices primarily designed to circumvent a TPM that *controls access* to a work, in violation of section 1201(a)(2), and to circumvent a TPM that *protects a right of a copyright owner*, in violation of section 1201(b). On both counts of trafficking in circumvention devices, the court finds Divineo liable and awards damages to the amount of $9 541 600.[927]

Divineo's primary defence to both contentions rested on the contention that the HDLoader program and the mod chips were not *primarily* designed to facilitate circumvention, and are therefore not within the ambit of either sections, because it could be used for other legitimate purposes. These include the ability to execute games on a PS console that were not manufactured by Sony (an interoperability argument), testing non-PS games on the console, reducing the load time of PS games by copying the games to external hard drive storage and eliminating the need to change game discs when a user chooses to play a different game.[928] In addition, the mod chips are the only way a user could play an imported PS game on a PS console restricted to the United States (a geo-blocking circumvention argument).[929]

[925] *Divineo* at 959.

[926] At 959.

[927] *Divineo* at 968. The award of damages was made in two parts payable by Divinio and others, including Divinio UK and Mr Legualt, the sole shareholder and president of Divinio Inc, personally.

[928] *Divineo* at 961.

[929] At 965.

The court agrees that the evidence supports Divineo's contention that the HDLoader and mod chips may be used for legitimate and lawful purposes, but rejects the contention that it is sufficient to show that the devices are not primarily designed or intended to be used for circumvention.[930] In support of its finding, the court states:

> "Downstream customers' lawful or fair use of circumvention devices does not relieve [Divineo] from liability for trafficking in such devices under the DMCA."[931]

This places the interoperability argument, and, by extension the reverse engineering exception for interoperability, in its appropriate context. It clarifies that the reverse engineering exception may not be used to contend that the making of, or trafficking in, circumvention devices may be justified because it facilitates interoperability.

In other words, it corrects the argument in *Davidson*. While the court in that case was wrong in its interpretation of the reverse-engineering exception,[932] it was correct in finding that the exception may not be used to argue that interoperability (by the user of the device) is a defence to the trafficking violations.[933]

The cumulative effect of *Davidson* and *Divineo* is, thus, as follows. The reverse-engineering exception permits the circumvention of a TPM in order to decompile, or otherwise inspect and analyse, a computer program if it is necessary to do so in order to create a new work that would be interoperable with the first program. The proviso, in the fourth factor, is that the act of decompilation may not be otherwise prohibited (by, for example, a license agreement). The proviso *does not* mean that the act of decompilation must itself be a non-circumvention act – in this regard the *Davidson* judgment has been corrected.

Furthermore, insofar as interoperability is concerned, the effect of the DMCA is clarified. It means that the *creation* of a derivative work, created by decompilation and

---

[930] This finding includes a weighing of the second factor in §1201(a)(2)(B) and/or §1201(b)(1)(B), that the device has "only limited commercially significant purpose or use other than to circumvent."

[931] *Divineo* at 965.

[932] See the analysis in para 4 2 4 1 2 2 above and the sources in fn 916 and 921.

[933] See *Divineo* at 964-5, with reference to preceding authority.

for the purpose of interoperability, will be a violation of sections 1201(a)(2) and/or 1201(b) if that work facilitates circumvention of a TPM. This will be true regardless of the fact that the actual circumvention of a TPM is not carried out by the party who created the work or device, but by the downstream user thereof. Consequently, the interoperability defence is not available to the creator of a circumvention device. This interpretation is expressly supported by three other judgments.[934]

This clearly places a significant restriction on the fair use decompilation exception. Prior to the DMCA, it was permissible to access a computer program **for any legitimate purpose** and create any derivative work regardless of its intended purpose. This position persists, after the DMCA, only in relation to computer programs that are *not protected* by a TPM or programs that *do not operate* as a TPM.

For all other situations, after the DMCA, it is only possible to legally access the code of a program if it is done for the **sole purpose of interoperability** and the derivative work is not primarily designed to be used as a circumvention device.

As a result, the DMCA has effectively narrowed the fair use decompilation exception to a single purpose, namely, interoperability, *and* limited the type of derivative work that may be created. There is no precedent in copyright law that prescribes *what* the purpose of an original work must be in order to avoid infringement. In this regard, the DMCA introduced an anomaly. In *MGM* the court finds that this limitation on the allowable function of a computer program is justified, not by copyright law, but by a purposive reading of the first amendment (the right to freedom of speech) which, under the circumstances introduced by the DMCA, imposes a lawful limitation on the right to

[934] See *321 Studios v Metro Goldwyn Mayer Studios Inc* (2004) 307 F. Supp. 2d 1085 (*MGM*) at 1097 (in relation to fair use reproduction of DVDs with the aid of a circumvention computer program); *Corley* at 443 (also in relation to the DeCSS circumvention technology and DVD content); *United States of America v Elcom Ltd* (2002) 203 F. Supp. 2d 1111 (*Elcom*) at 1130 (in relation to criminal DMCA infringement of the Adobe eBook Reader anti-circumvention measures and the freedom rights of the end-user).

free expression.[935] It finds support for this in both the *Elcom*[936] and *Corley*[937] decisions, where the same argument was advanced in support of the DMCA.

An examination of the constitutionality of the DMCA is outside the scope of this work. However, the potential impact of a limited decompilation exception on fundamental rights is directly applicable to this work and is examined later in the South African context.

4 2 4 1 3 DMCA exempted classes

To conclude the review of the development of the decompilation exceptions under American law, it is necessary to briefly consider the final balancing mechanism introduced by the DMCA in order to attempt a safeguard of the fair use rights.

The Librarian of Congress, acting on recommendations of the Register of Copyrights, is empowered by section 1201(2)(1)(C) to declare that certain particular classes of works are exempted from the application of the circumvention prohibition[938] for a period of three years from the date of publication of the exemptions.

The most recent list of classes of copyrighted work that are exempt was published in October 2018 and contain detail on the rulemaking procedure and the policy considerations that affected the decision, according to the factors the Librarian is obliged to consider in terms of the DMCA.[939]

---

[935] *321 Studios v Metro Goldwyn Mayer Studios Inc* at 1100-1.

[936] *Elcom* at 1128.

[937] *Corley* at 454.

[938] Section 1201(a)(1)(A).

[939] The factors, listed in section 1201(a)(1)(C)(i) to (v) are:

> '(i) the availability for use of copyrighted works;
>
> (ii) the availability for use of works for nonprofit archival, preservation, and educational purposes;
>
> (iii) the impact that the prohibition on the circumvention of technological measures applied to copyrighted works has on criticism, comment, news reporting, teaching, scholarship, or research;
>
> (iv) the effect of circumvention of technological measures on the market for or value of copyrighted works; and
>
> (v) such other factors as the Librarian considers appropriate."

The list of current exemptions[940] include 10 exempted types of use that fall within the general scope of this work, summarised by the Librarian of Congress as follows:[941]

- "Literary works distributed electronically (i.e., e-books), for use with assistive technologies for persons who are blind, visually impaired or have print disabilities
- Literary works consisting of compilations of data generated by implanted medical devices and corresponding personal monitoring systems
- Computer programs that operate the following types of devices, to allow connection of a new or used device to an alternative wireless network ("unlocking"):
    - Cellphones
    - Tablets
    - Mobile hotspots
    - Wearable devices (e.g., smartwatches)
- Computer programs that operate the following types of devices, to allow the device to interoperate with or to remove software applications ("jailbreaking"):
    - Smartphones
    - Tablets and other all-purpose mobile computing devices
    - Smart TVs
    - Voice assistant devices
- Computer programs that control motorized land vehicles, including farm equipment, for purposes of diagnosis, repair, or modification of the vehicle, including to access diagnostic data
- Computer programs that control smartphones, home appliances, or home systems, for diagnosis, maintenance, or repair of the device or system
- Computer programs for purposes of good-faith security research
- Computer programs other than video games, for the preservation of computer programs and computer program-dependent materials by libraries, archives, and museums
- Video games for which outside server support has been discontinued, to allow individual play by gamers and preservation of games by libraries, archives, and museums (as well as necessary jailbreaking of console computer code for preservation

---

[940] Librarian of Congress (2018) *Section 1201 Rulemaking: Seventh Triennial Proceeding to Determine Exemptions to the Prohibition on Circumvention* 6-8. The recommendations are codified as *Exemption to Prohibition on Circumvention of Copyright Protection Systems for Access Control Technologies* Federal Register Vol. 83 No. 208, 37 CFR Part 201 [Docket No. 2017–10].

[941] The full text of the exemptions, and the recommended regulatory text of each, are contained in the report at 9-16.

uses only), and preservation of discontinued video games that never required server support

- Computer programs that operate 3D printers, to allow use of alternative feedstock"[942]

The recommended regulatory text for each of the above exemptions is contained in the report. In light of this, it is clear that where decompilation is necessary, it will not be a violation of the DMCA if the TPM affixed to any of the above works is circumvented in the process. However, it should be noted that the above exemptions do not eliminate the interoperability limitation imposed by the DMCA reverse-engineering exception. In relation to the exempted classes of work, the reverse-engineering exception is simply replaced, until October 2021, by the wording of the relevant exemption, all of which contain inherent, but different, purpose limitations.

4 3 The United Kingdom position

Compared to the American position, the development of the decompilation exception in UK law is significantly more transparent. This is mainly due to the fact that jurisprudence on this point developed through statute, rather than case law, and is codified within the ambit of the doctrine of fair dealing, which does not rely on a case-by-case analysis, unlike the doctrine of fair use.

In light of the fact that the decompilation exception in South African law is likely to rely on a blended model of both fair use and fair dealing, with an emphasis on the first, it is necessary to review the UK position.

4 3 1 The founding principles for copyright protection in computer programs

The protection of computer programs as copyrightable work in UK law was introduced in 1985 by way of the Copyright (Computer Software) Amendment Act,[943] originally proposed as a private member's bill,[944] and gave effect to the founding principles

---

[942] Librarian of Congress *Section 1201 Rulemaking* 6-8.

[943] Copyright (Computer Software) Amendment Act UK ST 1985 c. 41 (the 1985 Amendment Act).

[944] See Bainbridge D I "The Copyright (Computer Software) Amendment Act (1985)" 1986 *Modern Law Review* 49 214 where the author states: "The new Act started life as a Private Member's Bill introduced by William Powell M.P. on December 5, 1984."

expressed in the report of the Whitford committee,[945] two subsequent green papers,[946] and the white paper.[947] In the first instance, it suggested that computer programs be accommodated under the act as a form of literary work despite "possible interpretive doubts"[948] about the appropriateness of this classification. The Whitford report indicated that, despite the absence of any supporting case law, computer programs were already protectable as a literary work[949] because it could be accommodated under the phrase "any similar process"[950] in the definition of "writing".[951]

Therefore, the report suggested only clarification of this in the 1985 amendment[952] to drive "home the fact that copyright **can subsist** in computer programs as literary works"[953] despite the precarious basis for this finding in copyright law which ties computer programs to copyright protection *only* by virtue of its physical appearance.

In the second and third instances, the founding documents dealt with "the largest stumbling block to copyright protection [which] was the fact that computer programs can be stored in a non-visible form and can be copied invisibly."[954] This led to the

---

[945] Whitford J N K 1977 *Copyright and Designs Law: Report of the Committee to Consider the Law on Copyright and Designs Cmnd. 6732* (*Whitford report*). See Dworking G "The Whitford Committee Report on Copyright and Designs Law" 1977 *Modern Law Review* 40 (6) 685 for a comprehensive review of the report.

[946] Reform of The Law Relating to Copyright, Designs and Performers' Protection Cmnd. 8302 (1981); Intellectual Property Rights and Innovation Cmnd. 9117 (1983). See DuCharme *Santa Clara High Technology Law Journal* 262 and Carty H and Hidkinson K "Copyright, Designs and Patents Act 1988" 1989 *Modern Law review* 52 (3) 369.

[947] Intellectual Property and Innovation Cmnd. 9712 (1986). See Bainbridge D I "Computers and Copyright" 1987 *Modern Law review* 50 (2) 202.

[948] Dworking *Modern Law Review* 699.

[949] DuCharme *Santa Clara High Technology Law Journal* 261-2.

[950] Section 48(1): "'writing' includes any form of notation, whether by hand or by printing, typewriting or any similar process."

[951] See DuCharme *Santa Clara High Technology Law Journal* 261 and footnotes for references to the Whitford report.

[952] Bainbridge 1986 *Modern Law Review* 215; DuCharme *Santa Clara High Technology Law Journal* 262.

[953] Bainbridge 1986 *Modern Law Review* 216.

[954] 216.

amendment of the meaning of "material form"[955] and an expansion of the meaning of "adaptation."[956] These founding principles were not unanimously agreed or welcomed by all. In particular, a minority of the committee expressed concern that the protection of computer programs as literary works, along with the extended meaning of adaptation, will constitute a restricted act in relation to the *use* of computer programs, which is "alien to fundamental aspects of copyright protection"[957] – in particular the idea/expression separation. Contemporary commentary on the 1985 amendment suggested that:

> "Computers, being an advanced form of technology with many unprecedented problems should be dealt with in quite separate legislation or, at least, in a separate category to fit the logical difficulties that the general copyright law is unable to cope with."[958]

However, the majority disagreed. It reasoned that copyright in computer programs should be as traditional as possible precisely because its peculiarities make it vulnerable to new and potentially infringing forms of use. Therefore, "if treated outside the ambit of general copyright law, there is no clear cut distinction between the idea and the expression"[959] which would leave the protected expression vulnerable and/or the unprotected ideas protected by some other means. While this reasoning is, in hindsight, unsatisfactory, it is understandable because the Whitford report was "a legally unsophisticated document"[960] and the 1985 amendment was intended to be an *interim*[961] solution, introduced under pressure,[962] until the 1988 Act[963] came into force.

---

[955] Copyright (Computer Software) Amendment Act section 2:

> "References in the Copyright Act 1956 to the reduction of any work to a material form, or to the reproduction of any work in a material form, shall include references to the storage of that work in a computer."

[956] Section 1(2).

[957] Dworking *Modern Law Review* 699.

[958] 699.

[959] Dworking *Modern Law Review* 699-700.

[960] 685.

[961] Bainbridge 1986 *Modern Law Review* 215.

[962] In particular the lobbying of the software industry pressure group the Federation Against Software Theft (FAST).

[963] Copyright, Designs and Patents Act 1988 Chapter 48 (*CDPA*).

As a result, the founding principles are expressed in the 1985 Amendment Act, which applied only to the previous Copyright Act,[964] as follows:

> "(1) The Copyright Act 1956 shall apply in relation to a computer program (including one made before the commencement of this Act) **as it applies in relation to a literary work** and shall so apply whether or not copyright would subsist in that program apart from this Act.
>
> (2) For the purposes of the application of the said Act of 1956 in relation to a computer program, a version of the program in which it is **converted into or out of a computer language or code**, or into a different computer language or code, **is an adaptation** of the program."[965]

4 3 2 The statutory position

The phrase "**as it applies**" to a literary work, in the 1985 amendment, and the lack of clarity provided by the founding documents on this point, left room for speculation about "why it was not treated as a literary work as such".[966] In 1988 the CDPA removed any doubt on this point with the introduction of a new provision:

> "'literary work' means any work, other than a dramatic or musical work, **which is written**, spoken or sung, and accordingly **includes**—
>
> (a) a table or compilation other than a database
>
> (b) a **computer program**
>
> (c) preparatory design material for a computer program and
>
> (d) a database"[967]

The CDPA does not contain a definition of computer programs, ostensibly to "ensure that considerable scope would be left to the courts for flexible interpretation in a rapidly

[964] Copyright Act 1956 Chapter 74.

[965] Section 1 (emphasis added).

[966] Dworking G and Taylor R D *Blackstone's Guide to the Copyright, Designs and Patents Act 1988: The Law of Copyright and Related Rights* 1ed (1989) 182. See the discussion above in relation to the writing analogy which, it is submitted, formed the basis for the classification of computer programs as literary work.

[967] CDPA section 3(1). The wording reproduced here is the current text and includes amendments made after 1988. The highlighted passages appeared in the original act in the same location.

developing technology,"[968] but it retains the close association between computer programs and writing as the basis for the classification as literary work.

Regarding the meaning of "adaptation" for purposes of infringement, the CDPA was amended by the Computer Programs Regulations of 1992[969] to provide that, in relation to a computer program, it "means an arrangement or altered version of the program **or a translation** of it,"[970] removing computer programs from the general meaning of adaptation.

In addition, the CDPA extended the term "translation" in relation to computer programs to include "a version of the program in which it is **converted** into or out of a computer language or code or into a **different computer language** or **code**."[971] The Computer Programs Regulations left this provision unchanged, except to delete the words "otherwise than incidentally in the course of running the program," which appeared at the end of the original text.

Lastly, the 1988 Act defined the infringing act of reproduction to include "storing the work **in any medium** by electronic means"[972] and "making of copies which are transient or are **incidental** to some other use of the work."[973]

These provisions make it clear that, because computer programs **are** literary works in terms of UK law, the process of decompilation will amount to infringement in, at least, two instances:

---

[968] Dworking and Taylor *Blackstone's Guide to the Copyright, Designs and Patents Act* 182. See also Lai S *Copyright Protection of Computer Software in the United Kingdom* 1ed (2000) 14 at 2.3 where the author repeats the same sentiment without citation, but provides HTLM language as an example. The accuracy of this submission is doubtful, considering that HTML is a language and copyright law clearly does not protect language.

[969] The Copyright (Computer Programs) Regulations 1992 No. 3233 (the *Computer Programs regulations*).

[970] CDPA section 21(3)(ab) (emphasis added).

[971] CDPA section 21(4).

[972] Section 17(2).

[973] Section 17(6).

Firstly, where it involves the making of temporary or incidental copies of the object code, such copies are deemed infringing copies. Second, when the process of decompilation delivers an approximation of the source code, it is deemed a translation in either one of two ways. If object and source code are considered different languages, decompilation amounts to a conversion into a *different language*. Alternatively, if object and source code are (correctly) understood to be an expression in the same language, decompilation will, nevertheless, constitute the making of a translation in a *different code*.

Since all forms of "translation" are, in relation to computer programs, considered a form of making an adaptation, the law places an express prohibition on decompilation in at least two ways.

The striking similarities between this position and that of South Africa (outlined above), are clear, including the concurrent reliance on the literary-works analogy in order to contort the meaning of adaptation to fit computer programs. Except for the addition of "notation" as a third alternative under South African law, the prohibition on decompilation rests on precisely the same provisions in law, with one notable difference - the classification of the type of work.

4 3 3 The Directives

The decompilation prohibition was, to a limited extent, lifted in 1992 following the Computer Programs Regulations which implemented the first Software Directive.[974] However, it should be noted that the amendments to the CDPA effected by the Computer Programs Regulations did not mirror the 1991 Software Directive.

---

[974] Council Directive on The Legal Protection of Computer Programs O.J. No. L122 91/250/EEC of 14 May 1991 (*1991 Software Directive* or *first Software Directive*). This directive was repealed in 2009 by the Directive of the European Parliament and of the Council on the Legal Protection of Computer Programs Directive 2009/24/EC (23 April 2009) *Official Journal of the European Union* 5.5.2009 L111/16 (*2009 Software Directive* or *second Software Directive*). Insofar as the decompilation prohibition is concerned (in Article 6 of both Directives), the wording is identical and no change to the CDPA was necessary.

The 1991 Software Directive suggest that:

> "The authorization of the rightholder shall not be required where **reproduction** of the code and **translation** of its form […] **are indispensable** to obtain the information necessary to achieve the **interoperability** of an independently created computer program with other programs."[975]

The Computer Programs Regulations introduced section 50B to the CDPA, which provides:

> "(1) It is not an infringement of copyright for a lawful user of a copy of a computer program expressed in a low level language -
> (a) to convert it into a version expressed in a higher level language, or
> (b) **incidentally** in the course of so converting the program, to **copy** it,
> (that is, to "decompile" it), provided that the conditions in subsection (2) are met.
> (2) The conditions are that—
> (a) it is **necessary** to decompile the program to obtain the information necessary to **create an independent program which can be operated with the program** decompiled or with another program ("the permitted objective"); and
> (b) the information so obtained is not used for any purpose other than the permitted objective."

Two notable differences between the 1991 Software Directive and the Computer Programs Regulations are the change from "indispensable" to "necessary" and the alternative wording for the permitted objective.

In order to position the decompilation exception within the established framework for literary works, it was necessary that computer programs be isolated from the fair dealing doctrine. For this reason, the Computer Programs Regulations also added section 29(4):

> "(4) It is not fair dealing -
> (a) to **convert** a computer program expressed in a low level language into a version expressed in a higher level language, or
> (b) incidentally in the course of so converting the program, to **copy** it,

[975] Article 6(1) of both Directives, subject to certain identical conditions listed in both Directives.

> (these acts being permitted if done in accordance with section 50B (decompilation))."[976]

In so doing, the CDPA removed the possibility to decompile a computer program for any existing fair dealing purpose other than interoperability, effectively importing the same inherent, and mistaken,[977] limitation created by the American courts after *Sega*. As a result, case law in the UK never dealt with the appropriate scope of computer programs in relation to decompilation and never considered the correct balance of the public and private interests or the functional nature of the work in relation to the scope of protection.[978]

This caused all subsequent case law in relation to the nature of computer programs to focus on the idea/expression dichotomy as an infringement question regarding non-literal elements, rather than a subsistence question in the first place. This is not surprising, considering that UK courts have been reluctant to embrace the idea/expression dichotomy.[979]

It is worth repeating that, where the nature of copyright in computer programs is concerned, "the discussion often runs the issues of protected subject matter and infringement together"[980] and that this work is only concerned with that part of the

---

[976] Section 29(4).

[977] See the analysis at fn 882 above. See also sources in fn 979 below.

[978] See the discussions of the decisions in *Mars U.K. Ltd v Teknowledge* Ltd [2000] E.C.D.R. 99 and *Cantor Fitzgerald International v Tradition (UK) Ltd* [2000] R.P.C. 95 in paragraphs 4 3 4 5 and 4 3 4 6 below.

[979] Caddick N, et al. *Copinger & Scone James on Copyright* 17ed (2019) para 9-228 with reference to the judgment in *IBCOS Computers Ltd. and Another v Barclays Mercantile Highland Finance Ltd. and Others* [1994] F.S.R. 275, discussed in paragraph 4 3 4 6 below. See also para 7-13, where the authors point out:

> "As with all such general statements of principle, however, the principle must be treated with caution and not taken too far. It is not a correct statement of English law that because a copyright work contains the expression of an idea it may be copied."

See also *John Richardson Computers Limited v Flanders and Another* [1993] F.S.R. 497 at 523 where the court points to the danger of over-reliance on this doctrine in UK law.

[980] Caddick, et al. *Copinger & Scone James on Copyright* para 7-83.

analysis that determine the scope of protection (the idea/expression dichotomy) where it is influenced by the prohibition on decompilation. In this sense, it straddles both prongs outlined in Copinger & Scone James, but it does not deal with the second infringement question, namely, the protection of non-literal elements or any other argument about the idea/expression dichotomy.[981]

This is made clear by the conditions for lawful decompilation, set by the Computer Programs Regulations:

> "(3) In particular, the conditions in subsection (2) are not met if the lawful user—
>
> (a) has readily available to him the information necessary to achieve the permitted objective;
>
> (b) does not **confine the decompiling to such acts as are necessary** to achieve the permitted objective;
>
> (c) supplies the information obtained by the decompiling to any person to whom it is not necessary to supply it in order to achieve the permitted objective; or
>
> (d) **uses the information to create a program which is substantially similar** in its expression to the program decompiled or to do any act restricted by copyright."[982]

Considering that, despite the decompilation exception in section 50B, a reproduction of a substantial part of the code will nevertheless amount to an infringing reproduction, adaptation or translation, there is no need for section 50B(3)(d) to exist. However, the inclusion thereof in this decompilation exception means that, if it is shown that the derivative work is infringing, the act of decompilation becomes, *retroactively*, an act of making an infringing adaptation of the original work. This effect is not only superfluous but also unhelpful. It serves only to enforce the interoperability limitation.[983]

---

[981] See paragraph 3-62 where the authors point out:

> "Nevertheless because the question is complex, and the discussion often runs the questions of protected subject matter, originality and infringement together, it may be helpful to say something about it here. In simple cases, where for example some or all of the source code can be seen to have been literally copied, the nature of the protected subject matter is not in doubt. Problems that have arisen are mainly concerned with allegations of non-literal copying, and whether the subject matter of a copyright in such works extends to such things as the functionality of a computer program or programming languages."

[982] Section 50B(3) (emphasis added). The wording of the conditions is based on, but not identical to, the conditions prescribed by the Software Directives.

[983] Some have suggested that subsection (d) should be read in light of the second Software Directive, to the extent that:

In light of the fact that the CDPA does not, like the US Copyright Act, allow for the *ad hoc* development of further legitimate uses, the introduction of section 50C[984] does not temper the limitation on decompilation or permit decompilation for any other purpose. A further limitation was placed on the act of decompilation with the introduction of the 2003 Copyright Regulations,[985] which implemented the InfoSoc Directive.[986] Prior to this, section 296 of the CDPA consisted of a limited restriction on the manufacture and dealing in devices that are "specifically designed or adapted to" circumvent *copy*-protection measures applied by the copyright owner, and the publication of information that "intended to enable or assist" circumvention.[987]

The 2003 Copyright Regulations replaced the whole of section 296 and split the anti-circumvention protection measures[988] between computer programs[989] and all other works.[990]

> "It seems likely that courts in the UK will interpret the words 'which is substantially similar in expression' as being equivalent to 'which reproduces a substantial part'." See Caddick, et al. *Copinger & Scone James on Copyright* para 9-228.

[984] Section 50C:

> "(1) It is not an infringement of copyright for a lawful user of a copy of a computer program to copy or adapt it, provided that the copying or adapting—
>
> (a) is **necessary for his lawful use**; and
>
> (b) is not prohibited under any term or condition of an agreement regulating the circumstances in which his use is lawful."

[985] The Copyright and Related Rights Regulations 2003 2003 No. 2498 (*2003 Copyright Regulations*).

[986] Directive of the European Parliament and of the Council on the Harmonisation of Certain Aspects of Copyright and Related Rights in The Information Society Directive 2001/29/EC of 22 May 2011 (*InfoSoc Directive*).

[987] See the original Queen's Printer version of the CDPA, part VII, section 296.

[988] See *Kabushiki Kaisha Sony Computer Entertainment Inc. and Others v Ball and Others* [2004] EWHC 1738 (*Kabushiki*) at 249-250:

> "In effect the anti-copy-protection provisions are now split into two groups. New s.296 is restricted to anti-copy-protection devices which allow copyright computer programs to be copied. s.296A onwards contains similar, but not identical, provisions to counter anti-copy-protection devices designed to facilitate the copying of other types of copyright work."

[989] Section 296.

[990] Section 296ZA.

In relation to computer programs, the 2003 Copyright Regulations provide that where a technical device, "intended to prevent or restrict acts that are not authorised by the copyright owner of that computer program and are restricted by copyright",[991] has been applied to a computer program, it shall be an infringement to deal[992] in "any means the sole intended purpose of which is to facilitate the unauthorised removal or circumvention of the technical device"[993] or to publish information pertaining to the circumvention.[994]

Under these circumstances, the form of infringement is described as "the same rights […] as a copyright owner has in respect of an infringement of copyright"[995] and are available to the copyright owner, publisher, distributor or exclusive licensee of the computer program protected by the anti-circumvention device[996] *and* the intellectual property rights owner or exclusive licensee of the anti-circumvention device.[997] The rights are conferred on these parties individually and concurrently.[998]

Unlike the DMCA, the CDPA does not prohibit the act of circumventing the protection measure applied to a computer program, although it does prohibit the act in relation to all other types of work.[999] Therefore, in the case of technologically-protected computer programs, the CDPA does not impose an *access* restriction. This has avoided the unintended and overbroad extension of copyright protection, and subsequent constitutionality challenges, created by the DMCA.

The clear intention of section 296, as expressed in case law, is to restrict the trafficking in circumvention devices that could be used to overcome technical protection to

---

[991] Section 296(6).

[992] The word "deal" is used here to summarise the list of acts described in section 296(1)(b)(1):

> "manufactures for sale or hire, imports, distributes, sells or lets for hire, offers or exposes for sale or hire, advertises for sale or hire or has in his possession for commercial purposes"

[993] Section 296(1)(b)(1).

[994] Section 296(1)(b)(ii).

[995] Section 296(2).

[996] Section 296(a) and 296(b).

[997] Section 296(c).

[998] Section 296(3).

[999] See section 296ZA et seq.

computer programs.[1000] It is not supposed to be applicable beyond this point as, for example, an alternative form of infringement where a work has not been reproduced. However, section 296 applies only where the act of circumvention or removal of the protection measure has not been authorised[1001] *and* the technical protection measure is applied in order to prevent or restrict any act that is not authorised by copyright law **or the copyright owner**.[1002]

Thus, where an act, such as making a derivative work, is expressly **unauthorised** by, for example the terms of a license, it will be an infringement under section 296 to make such a derivative work. This will be true in all cases, unless the authorisation is expressly granted by copyright law,[1003] in which case it will override the restriction imposed by the license. Considering that section 50 only grants authorisation for certain specific purposes, section 296 makes it possible to prevent the creation and distribution of computer programs that would, in the normal course of events, not amount to copyright infringement.

The case of decompilation is a perfect example of this. Section 50B is an instance where an act is expressly authorised, namely, decompilation. Thus, if a technological measure prevents decompilation, that measure may be lawfully circumvented because it is authorised by copyright law. Any contractual provision to the contrary would be void.[1004] In relation to computer programs, the list of authorised acts, which may not be excluded by contract, are:

> "(a) the making of any back up copy of the program which it is necessary for him to have for the purposes of the agreed use;

[1000] See *Kabushiki* at 248 where the court held that with the amendment, the legislator "wished to prohibit trade in devices which overcame copyright-protection without regard to whether that assisted copyright infringement." See also para 26 on page 250 the judgment where the court observes that this sentiment is applicable to the CDPA as amended, but not before.

[1001] Section 296(1)(b)(i) refer specifically to "**unauthorised** removal or circumvention of the technical device."

[1002] Section 296(6) refer to "acts that are **not authorised** by the copyright owner of that computer program and are restricted by copyright."

[1003] Section 296(6).

[1004] Section 50B(4).

> (b) **where the conditions in section 50B(2) are met**, the decompiling of the program; or
> (c) the observing, studying or testing of the functioning of the program in accordance with section 50BA."[1005]

However, decompilation is expressly limited by the interoperability requirement. Therefore, as the above section makes clear, decompilation for any other purpose is not authorised by copyright law and, thus, may be excluded by contract – which means it is unauthorised for purposes of section 296. Consequently, section 296 remains applicable to decompilation. In other words, a licensor is free to prohibit all access to code in general and rely on section 296 as a form of infringement, by circumvention of the measure it has imposed to prevent access, where access has been obtained, and a program created, for a purpose that does not fall within the meaning of interoperability. Therefore, section 296 does create an alternative means of liability, at least in cases of decompilation.

The facts in the *Kabushiki* case are a good illustration of this. The contentious circumvention device in this case, the MESSIAH2 chip, was designed to allow, inter alia, that Sony PlayStation games licensed for one region of the world would be compatible with PlayStation consoles and televisions in another region of the world by rendering ineffective the technical protection measures imposed by Sony on its consoles and PS game discs.[1006] Sony alleged that the manufacture and distribution of the chip amounted to an infringement in terms of section 296[1007] before and after the amendment. The court found that the protection measures applied by Sony fell within the meaning of the Act and that the conduct of the defendant, Ball, amounted to infringement because:

> "Sony's copy-protection system is designed to prevent all uses of copied PS2 game CDs and DVDs on PS2 consoles and all uses of non-PAL games on PAL consoles (and vice versa), The Messiah2 chips' sole purpose is to circumvent this."[1008]

[1005] Section 296A (emphasis added).

[1006] *Kabushiki* at 243-4.

[1007] *Kabushiki* at 244.

[1008] At 252.

Since regional or device compatibility is clearly not a form of interoperability within the meaning of section 50, it is impossible to argue that the device manufactured by Ball is not prohibited by section 296 because Sony's technical measures do not protect a right authorised by law. The court expressly denied this form of defence when it found that "this argument could only succeed if either of these classes of activities were authorised by Sony."[1009] In other words, if the Sony license or the law permitted the circumvention for purposes of removing the region controls, the device would be lawful. But, because the Sony technical measure effectively restrict regional control circumvention, and copyright law does not allow such measures to be circumvented (because it is not a permitted form of interoperability), the device is prohibited by section 296 and a case for infringement had been made. It is not necessary to show that the law also expressly prohibits the conduct, all that is required is that the prohibition imposed by the licensee rests on an exclusive right granted by copyright. In this case, the region control measures relied on the exclusive reproduction and distribution rights and circumvention thereof was prohibited by Sony through the imposition of a technical protection measure.

The inevitable result is that the CDPA is much closer to the DMCA than a first reading of the Act would suggest, and has created an alternative form of infringement, in addition to infringement under the rest of the CDPA, in the case computer programs. In the context of this work, the effect is that, while decompilation is permitted for the purpose of interoperability, section 296 has created an additional obstacle to access to computer code and the *manner* in which that code may be used, even where no protected elements are reproduced.

In short, if a computer program is decompiled to access the code and it is used, but not copied in any way, to manufacture (program) a new program that would interact with the original or another program, it is not copyright infringement by operation of section 50B. However, if that program could be used as a device to overcome technical protection measures, the manufacture (decompilation and subsequent new programming) of that program is, nevertheless, an infringement in terms of section 296. The negative effect of this is somewhat tempered by the "sole intended

[1009] At 250.

purpose"[1010] factor, discussed further below, but it does not alleviate the burden it places on programmers who wish to decompile a program in order to create something new where that work is not made for the strict purpose of interoperability.

Some noted scholars have recognised this effect as a "tension between technical devices on the one hand and the permitted acts on the other"[1011] where, for example, copyright implicitly permit an act to be carried out in relation to a work, such as, making back-up copies or amendments to the work for whatever use purpose, but the licensor has prohibited that activity and put a technical measure in place. As the authors note, there is *in theory* no tension because section 296 does not prevent the act of circumvention, only the act of making devices that could be used for circumvention. However, as the authors submit, *in practice* a tension is created because "it is impossible for a supplier to know exactly what will be done with the means or information he supplies. This may well deter suppliers from supplying those who genuinely wish only to carry out permitted acts."[1012]

It has been shown above that the authors are correct. However, it should be noted that this tension extends beyond that which the authors recognise, insofar as section 296 not only discourages the supplier from distributing the work, it also discourages the *making* of such work. This does not only limit the utility of the decompilation exception, it also, crucially, established a wholly new means to monopolise the use of computer programs and the ideas contained therein. As the authors point out, "it is doubtful whether this is consistent with the requirement of art.7(1) of the Software Directive that the rights in respect of technical devices should be 'without prejudice to' the permitted acts."[1013]

This question came before the Court of Justice of the European Union regarding the situation where a technological protection measure may not be circumvented because it simultaneously restricts acts that amount to copyright infringement and acts that are

---

[1010] Section 296(1)(b)(i).

[1011] Caddick, et al. *Copinger & Scone James on Copyright* para 15-17.

[1012] Paragraph 15-17.

[1013] Caddick, et al. *Copinger & Scone James on Copyright* para 15-17.

otherwise lawful. In such cases, the court suggested that national courts should address the tension described above by considering whether the copyright owner could have imposed a less restrictive technological measure by considering "the relative costs of different types of technological measures, of technological and practical aspects of their implementation and a comparison of the effectiveness of those different types of technological measures as regards the protection of the rightholder's rights, that effectiveness however not having to be absolute."[1014]

In addition to the heavy evidentiary burden this would place on parties, the court also suggests that the "evidence of use which third parties actually make"[1015] of the allegedly circumventing device should be considered, in particular "how often those devices, products or components are in fact used in disregard of copyright and how often they are used for purposes which do not infringe copyright."[1016]

It is submitted that this judgment is not only unhelpful in balancing section 296 in relation to legitimate uses but also counter-productive. It places a reverse burden of proof on the programmer to collect and submit evidence from third parties about their subjective intention when obtaining or using the computer program, in order to argue that the program is not a circumvention device in terms of the Act. This preponderance of usage requirement is entirely contrary to copyright law and creates a test, based on empirical evidence, before an individual is entitled to act in terms of an existing fair dealing exception merely because a licensor decided to make that act technically difficult.

Regarding the nature of the prohibited circumvention devices, the new wording of section 296 differs from the DMCA insofar as the former refers to devices which have the "sole intended purpose"[1017] of circumvention, while the latter refers to devices "primarily designed or produced"[1018] for the purpose of circumvention. The previous

---

[1014] *Nintendo Co. Ltd v PC Box Srl* Case C-355/12 [2014] E.C.D.R. 6 at 38.

[1015] *Nintendo Co. Ltd v PC Box Srl* at 38.

[1016] At 38.

[1017] Section 296(1)(b)(i).

[1018] DMCA section 1201(a)(2)(A) and section 1201(b)(A).

version referred only to "any device or means specifically designed or adapted to circumvent the form of copy-protection employed".[1019] In addition, the amended section 296 is limited to persons who deal in such devices "knowing or having reason to believe that it will be used to make infringing copies."[1020] The introduction of this element in the CDPA has raised questions of certainty regarding the identity of the party to whom the state of mind must be attributed,[1021] creating the possibility of a counter-argument based on the ignorance of a downstream distributor or marketer of circumvention devices. Insofar as it has been suggested that this is an incorrect implementation of article 7 the InfoSoc Directive,[1022] which does not contain a knowledge requirement, and must therefore be read out of the CDPA, it is submitted that this view is not correct.

It is clear from case law[1023] that the knowledge requirement in the CDPA must be met and is general in nature[1024] and thus applicable to the state of mind of the alleged infringer, whether or not that party is the infringer. The court's interpretation of this requirement has, however, made it clear that the knowledge need not be specific to a particular program.[1025] In other words, the knowledge is about the nature of the device as a circumvention tool. It need not be shown that the infringing party also knew that it will be used to circumvent the protection in relation to the computer program in that specific case – general knowledge about the ability to circumvent is sufficient.

[1019] Section 296 before the 2003 Copyright Regulations came into effect.

[1020] Section 296(1)(b).

[1021] In other words, whether it is the intention of the manufacturer of the device that is determinative or if, in some cases, the intention of the party who deals in the device but is not the manufacturer. See Caddick, et al. *Copinger & Scone James on Copyright* para 15-15 where the authors submit that the test is objective and, therefore, refer to the intention of the infringing party in the case, whether or not they are the actual manufacturer of the device.

[1022] See Caddick, et al. *Copinger & Scone James on Copyright* para 15-14.

[1023] *Nintendo Company Ltd v Playables Ltd, Wai Dat Chan* [2010] EWHC 1932 (Ch).

[1024] At 292-3 the court held:

> "I shall proceed on the basis of the language of the statute rather than the Directive, which, after all, only directs member states to provide "appropriate" remedies."

[1025] At 293 para 35.

4 3 4 Case law

Following the very early, compared to the American position, introduction of a statutory decompilation exception, there exists no UK case law on the legality of the act of decompilation. However, some judgments on other copyright issues did have an impact on the development of the statutory exception or shed some light on the literary-analogy that underpins UK law in relation to computer programs. These cases are considered below with a focus on decompilation and interoperability.

4 3 4 1 *Sega Enterprises v Richards (1983)*[1026]

This is one of the earliest cases to deal with computer programs under UK law and, except for one other,[1027] the first case where reverse engineering (in the form of disassembly) of the code is mentioned. Because this matter was an interlocutory motion for an injunction, and dealt with the law under the 1956 act before the 1985 amendments, the judgment was delivered only in relation to one question: can copyright subsist in a computer program?

The work in question was the video game FROGGER, recorded on a memory chip attached to a logic board and intended for use in an "electronic apparatus,"[1028] i.e the traditional arcade consoles. Richards produced and sold "conversion kits for setting up an imitation of "Frogger" on machines belonging to their customers."[1029] The conversion kit consisted of a logic board and the necessary memory devices, on which an imitation of the game was recorded by Richards.

[1026] *Sega Enterprises Limited v Richards and Another* [1983] F.S.R. 73 (*Richards*).

[1027] *Gates and Others v Swift and Others* [1982] R.P.C. 339. In this case, the infringing programs were recorded on cassette tapes obtained without license and altered before resale. On the facts of the case no disassembly took place. However, the court (at 339) remarks in passing that reverse engineering is in principle possible where the work is distributed in hexadecimal code: "The assembly language could be recovered from the object code by means of a special program called a disassembler."

[1028] *Richards* at 74.

[1029] At 74.

Richards admitted that their program was “based on [the] machine code program” belonging to Sega, but contended that this does not amount to copyright infringement because “in English law there is no such thing as copyright in a computer program”.[1030]

Richards argued that “because of the very large part played by automatic means, that is to say by the computer itself, in the subsequent development of the program”,[1031] copyright protection in a computer program does not extend any further than the written flowcharts and other preparatory material.

Considering that the game, as distributed by both Sega and Richards, was recorded in machine code (object code), and Richards admitted to using the machine code to create their version, it is clear that decompilation must have occurred and, after that, the source code work was compiled into a new program. In fact, Richards indicated that “considerable work” was done on the Sega object code in order to create the new program.

The court does not conduct a factual or legal analysis, but concludes only that, as a general proposition, copyright protection subsists in the assembly language (source code) version of the work as a literary work.[1032]

Therefore:

> “The machine code program **derived from it** by the operation of part of the system of the computer called the assembler is to be regarded, I think, as either **a reproduction or an adaptation of the assembly code program**, and accordingly for the purposes of deciding this motion I find that copyright does subsist in the program.”[1033]

[1030] At 74.

[1031] At 75.

[1032] At 75.

[1033] *Richards* at 75.

4 3 4 2 *Thrustcode Limited v W.W. Computing (1983)*[1034]

This is the first case where the protection of computer programs under copyright law is confirmed at trial. The case dealt with several computer programs that improved the allocation of work in large factories[1035] and the alleged infringement in relation to competing programs made by the same programmers.

The case did not involve any disassembly or decompilation, and was dismissed because the court could not determine infringement without evidence of copying being provided in the form of printed source code.[1036] Therefore, an analysis of the facts of this case is not helpful to this work.

However, the manner in which the court finds that copyright law is applicable to computer programs is important to this work. The court relies on the limited and strictly reserved opinion of the court in Richards to assume it "should proceed on the footing that literary copyright is capable of subsisting in a computer program".[1037]

For this reason, it held that a case of infringement must be decided on the text of the program and, where it is not present, a decision of copying would be difficult[1038] to make because "you must look at what the programs are, and not merely at what they do or can do."[1039]

In this context, the court's finding that "in broad terms, literary copyright subsists not so much in ideas as in the expression of ideas"[1040] is not merely a general copyright sentiment but a finding on the nature of computer programs as literary works and the

---

[1034] *Thrustcode Limited and Another v W.W. Computing Limited* [1983] F.S.R. 502 (*Thrustcode*)

[1035] At 503.

[1036] At 507. The claimant submitted into evidence only a small abstract (75 lines) of source code.

[1037] At 505.

[1038] The court (at 505) indicates that it would be difficult but not be impossible under such circumstances:

> "In some cases, no doubt, it may be possible in some other way to demonstrate that one is a copy of the other, as where there is some evidence or some admission that when one computer was being programmed, someone was watching and was programming a rival computer in the same or a similar way."

[1039] *Thrustcode* at 506.

[1040] At 507.

manner in which infringement must be determined. It makes it clear that, because programs are literary works, there must be evidence of copying of the text. Consequently, where there is evidence of copying of the text, infringement *per se* has been established.

This makes it easy to see why the law has developed in two different directions simultaneously from this basic observation. In the first instance, it has given rise to the contentious protection of non-literal elements (where there is no evidence of text copying). In the second instance, it has positioned the law in relation to computer programs so close to literary works that the decompilation prohibition was an inevitability.

Together, the *Richards* and *Thrustcode* decisions represent the clearest illustration of the origins of the literary-analogy that has plagued copyright in computer programs ever since. These cases dealt with computer programs as literary works from the outset and introduced the concepts of adaptation and translation to computer programs.

It is also, in all likelihood, the only two cases upon which the Whitford Report could have based its assumption that the 1956 Act protects computer programs as literary works. The findings of the court in these cases are echoed by the reliance on the "written" characteristic of computer programs in the Whitford Report.

4 3 4 3 *Total Information Processing Systems v Daman Limited (1992)*[1041]

The origin of the interoperability limitation inherent in UK statutory development has an equally oversimplified origin.

In this case, Daman attempted to restrain the plaintiff (TIPS) from creating or distributing any code that allows its program, CONTROLLER20 program, to interact with the defendant's, entitled DAMANPAY. Both suites of programmes dealt with "contract costing, subcontractor ledger or job estimating"[1042] in the engineering sector,

[1041] *Total Information Processing Systems Limited v Daman Limited* [1992] F.S.R. 171 (*Daman*).

[1042] At 176.

and required that data regarding cost factors be communicated from one constituent part of the program to another. The dispute arose when TIPS offered for sale the CONTROLLER20 program which could receive and process data form one or more parts of the DAMIANPAY suite of programs.

The court dismissed the copyright infringement claim because it found that the portion of code that facilitates the interaction between the various programs in both suites, are not subject to copyright protection. In other words, where the programs relate only to interface specifications, no copyright exists.

The courts finding is primarily concerned with the fact that the contentious program in the DAMIANPAY suite was itself "specifically devised and designed to operate with other programs"[1043] and, therefore, it cannot claim copyright in that part which is responsible for its interoperability.

The court states that in general, as on the facts of this case, "many programs are **written with the intention** that they be **interfaced** with other programs."[1044] This is the primary substantive legal reason[1045] why the court decided that copyright does not vest in interface specifications, even though they are represented in source code. The merit of this finding need not be examined, because it leads only to a discussion of the idea/expression dichotomy in relation to the merger doctrine, which is outside the scope of this work.

However, as far as the decompilation prohibition is concerned, it reveals the early thinking of the court and, by extension, lawmakers, on the need to address the scope of protection by excluding those aspects that relate to interoperability.[1046] This is, therefore, the origin of the interoperability limitation in the decompilation exception in

[1043] At 177.

[1044] At 181.

[1045] The court (at 181) provides two other reasons: First, that the interface specifications could only be expressed in one way and is therefore not protectable expression. Second, that steps were taken to secure the confidentiality of the source code, which indicate that the work was not otherwise protectable.

[1046] See Caddick, et al. *Copinger & Scone James on Copyright* para 7-89 for a discussion of the exclusion of interface specifications under article 6 of the Software Directive.

UK law.[1047] It is instructive to note that the basis of the limitation is inward-facing. It considered the nature of the work as inherently reliant on interoperability. It is not outward-facing, insofar as it was not developed to address the need of others to inspect and learn from the work.

In this light it is clear to see why the interoperability limitation exists and why it does not consider the plethora of other justifiable reasons why a user may wish to decompile the program.

4 3 4 4 *John Richardson Computers Limited v Flanders (1993)*[1048]

In this case, the dispute related to the reproduction of the structure and arrangement of coded instructions in relation to two competing programs for the production and printing of pharmaceutical labels.[1049]

Evidence in the case made it clear that the contentious work was created with the aid of disassembled or decompiled code because some of the similarities between the work would otherwise be "a work of genius"[1050] and the result of "quite extraordinary feats of memory and unaided mental structural analysis and design."[1051] Despite this, the court elects to believe the testimony of the creator of these works that he had no access to the source code and therefore did not copy any instructions from the original program.[1052] Consequently, the judgment contains no indication of the relevance of decompilation to a finding of infringement.

The two programs, namely, the BBC program of Richardson, and the PHARM-ASSIST and CHEMTEC programs created by Flanders, were written in different programming languages but performed all of the same functions with minor variations in the output

---

[1047] Section 50B.

[1048] *John Richardson Computers Limited v Flanders and Another* [1993] F.S.R. 497 (Flanders).

[1049] The facts of the case, in particular in relation to the ownership and licensing of the original programs, and the past employment relationship between some of the parties, is very complex and will not be repeated or summarized here. It has no impact on the analysis of this case for purpose of this work.

[1050] *Flanders* at 546.

[1051] *Flanders* at 546.

[1052] At 548.

of each. All substantial differences between the works were as a result of the technical specifications and limitations of computer language, and processing capacity, of the different computers for which the individual programs were developed.

After a protracted analysis of the constituent *functions* and *operation* of the programs, rather than a comparison of the code,[1053] the court determines that some similarities indicate copying of protected expression and that these are collectively a substantial part of the original.[1054] Much may be made of the court's methodology in this case, but it will only be relevant to a general idea/expression discussion and not to decompilation.

However, the court also deals with the alternative contention that, in this case, the varied and individually minor instances of copying are substantial when a computer program is treated as a compilation of literary works. The argument is that the structure of a computer program and its constituent functional parts is deserving of protection in the same manner as the selection and arrangement of elements in a compilation.[1055] Although the court, correctly, finds the application of literary compilations to computer programs difficult because "the work and skill of the writer of the original program lies in the design of the component parts rather than in the selection of particular components,"[1056] it does not dismiss the idea.

In fact, the court indicates that it may be entertained in another case where it is "possible to compare code with code at some level, but that has not been so in this case"[1057] because the programs were written in different programming languages.

There is no point in discussing the possible protection of computer programs as a compilation. It has been adequately dismissed by several other courts[1058] because

---

[1053] At 549-558.

[1054] At 558.

[1055] At 558.

[1056] *Flanders* at 559.

[1057] At 559.

[1058] *Navitaire Inc v easyJet Airline Co Ltd* [2004] EWHC 1725 (Ch); *Forensic Telecommunications Services Ltd v Chief Constable of West Yorkshire* [2011] EWHC 2892 (Ch). For a discussion of this there was "no compilation of commands, only an 'accretion' by virtue of their individual formulation."[1059]

However, this judgment is important to this work because it emphasises the extent to which computer programs were equated with literary works and, consequently, influenced the development of the decompilation prohibition. It introduced the notion that, even in the absence of textual reproduction, computer programs must be protected against structural borrowing to, at least, the same extent as literary compilations. Therefore, where the work is decompiled in order to copy its structure for a purpose that is not interoperability but, for example, efficiency or expedience, copyright law must be applied to prevent this *because*, in its purest form, computer programs as literary works are deserving of protection against textual copying *and* reproduction of the labour and skill invested in the arrangement of the work.

The extent to which the literary-analogy is embedded in the construction of copyright law in relation to computer programs is evidenced by the fact that more recent case law has repeated the likelihood that computer programs are capable of protection as a literary compilation,[1060] although opinion on this point is still divided.[1061]

---

point, and the possibility of protecting computer programs as compilations, see Caddick, et al. *Copinger & Scone James on Copyright* para 3-52 and the sources listed in fn240.

[1059] Caddick, et al. *Copinger & Scone James on Copyright* para 3-52.

[1060] See *SAS Institute Inc v World Programming Ltd* [2013] EWCA Civ 1482 at 80 where the court states:

> "I can see no reason in principle why something that has grown by accretion should, for that reason alone, be deprived of copyright protection."

[1061] See *Navitaire Inc v easyJet Airline Co Ltd* at 346 [92] where the court held that "there is no overall compilation, but merely an accretion of commands" and, consequently, no originality in the arrangement because the order is pre-determined by the prescripts of the programming language and the application of **syntax**, variations in the names of the commands, is not sufficient to qualify for protection as a compilation.

4 3 4 5 *Mars UK Limited v Teknowledge (2000)*[1062]

The tension created by section 296 between decompilation of computer programs and fair dealing was at issue in this case and established that, under UK law, it cannot be addressed by the creation of additional fair dealing exceptions by means of case law.

In this matter, Teknowledge decompiled and copied parts[1063] of the CASHFLOW computer program and associated database, in which Mars held copyright, in order to create a new program that was capable of adapting the operation of the CASHFLOW program.

The CASHFLOW program was widely used as a security mechanism in a variety of coin-operated machines. The purpose of the program is to discern between the different coins inserted in a machine and check the authenticity thereof by a sophisticated measurement and comparison process to database information about a wide variety of available coins.[1064] The CASHFLOW program was recorded on an EPROM chip, which allowed it to be altered and re-programmed to accept new coins or reject others by amending the parameters, in the computer program and the data file, which a coin must meet.

Teknowledge is a reputed service provider in the business of re-programming a variety of coin-discriminator programs produced by other manufacturers and the Classic program produced by Mars before the CASHFLOW program was introduced. For commercial reasons, in order control the licensing of access to the code of the CASHFLOW program, Mars did not make the necessary information available which would enable Teknowledge to re-program the CASHFLOW program.[1065] This led to need to decompile the program in order to make the necessary changes when new coins were introduced.

[1062] *Mars U.K. Ltd v Teknowledge Ltd* [2000] E.C.D.R. 99 (*Mars*).

[1063] In particular the “serial communications protocol and an encryption system” and precise algorithms. See *Mars* at 102 [5].

[1064] *Mars* at 101.

[1065] *Mars* at 102.

Mars claimed that this amounted to copyright infringement: (1) when transient copies of the program were made during decompilation; (2) literal copying of portions of the code took place, which was necessary to effect re-programming; and, (3) the manufacture and distribution of a circumvention device in terms of section 296.[1066]

Since Technowledge's decompilation steps were not carried out for the purpose of interoperability within the strict limits of section 50B, and it also copied literal portions of the code that were necessary to perform the re-programming, it conceded to several instances of copyright infringement at the request of the court.[1067] This, in turn, allowed Mars to abandon the alternative claim of infringement based on section 296.

As a result, the case dealt with only two issues. In the first instance, a breach of confidence claim, which is not relevant to this work. In the second instance, an argument in defence of copyright infringement based on the *British Leyland*[1068] "right to repair"[1069] or common law spare-parts defence.

The court found that amendments to the 1988 CDPA abolished copyright protection for industrial designs[1070] and, along with it, the spare-parts defence.[1071] In relation to pre-1988 works, the defence has residual application.[1072] However, this defence is limited to works of a certain type, namely industrial designs. Where computer programs are concerned, the CDPA and the directives provide "a complete statutory code"[1073] of the applicable exceptions.

[1066] At 103 [9].

[1067] At 103-4.

[1068] *British Leyland v. Armstrong* [1986] A.C. 577.

[1069] *Mars* at 103[9]. See also Caddick, et al. *Copinger & Scone James on Copyright* par 5-244 to 5-246 for a discussion of the nature of this right.

[1070] Replaced by the "must match" and "must fit" exceptions in section 213(3)(b). See Caddick, et al. *Copinger & Scone James on Copyright* para 13-71.

[1071] *Mars* at 105 [14].

[1072] At 106 [15].

[1073] At 106 [17].

Therefore, the court agrees with Mars that "nothing in the [Computer Programs] Directive, and consequently in sections 50A–C, provides for any 'repair' or update exception."[1074] For this reason, the court refuses the defence because "it is not for national judge-made laws to override or add to what are clearly intended to be Community wide rules."[1075]

This makes is clear that the UK court is not free, unlike that of the US, to extend the scope of fair dealing or the decompilation exception at all, even where, as in this case, it is clearly for a justifiable and necessary purpose.[1076] As a result, the court found that *the act of decompilation* and the subsequent reproduction of coded portions amounted to separate instances of infringement.[1077]

This decision clearly illustrates the undue and unjustifiable limitation[1078] on the scope of decompilation imposed by the strict interoperability qualifier. It also shows that there are cases where copyright law in relation to computer programs fail to maintain an adequate balance between access to, and reuse of, ideas and the monopolisation of expression – a balance that the decompilation is intended to maintain and fails to do.

[1074] At 107 [18].

[1075] At 107 [18].

[1076] The court also considered the scope of the spare-parts defence and held that, even if it was wrong to find that the defence does not apply to computer programs, the actions of Teknowledge would not qualify for exemption under this defence because it went beyond the public policy considerations that underpin the defence insofar as Teknowledge sought to compete commercially with Mars regarding re-programming of the coin machines. See the discussion at 108 [23] and 109 [26]-[27].

[1077] See Caddick, et al. *Copinger & Scone James on Copyright* para 26-311 for a discussion of the case in relation to the infringement of the database rights.

[1078] See the discussions at fn 882, 977 and 979 above.

4 3 4 6 *IBCOS Computers Limited v Barclays Mercantile Highland Finance Limited (1994)*[1079] and *Cantor Fitzgerald International v Tradition UK Limited (2000)*[1080]

These two cases played an important role in the development of UK copyright jurisprudence on the protection of computer programs, primarily concerning non-literal elements. Both cases dealt predominantly with the questions of originality and substantiality in the context of infringement and, as such, made a significant contribution to UK law regarding the limited recognition of the idea/expression dichotomy itself and the test for infringement regarding functional elements.[1081] However, because both cases approached this question from the perspective of infringement, a comprehensive discussion of these cases fall outside the scope of this work because it adds nothing to the analysis of the idea/expression dichotomy in relation to decompilation. In other words, there is little to be learned from these cases about the legality of gaining access to code under the terms of section 50B or any other exception.

For the sake of completeness of this work, those aspects of these cases that are relevant to decompilation are described here.

In *Cantor*, the defendant, Tradition, engaged in extensive reverse engineering of the computer program owned by Cantor Fitzgerald International (CFI) in order to create a close approximation thereof. This process did not require decompilation, but Tradition admitted to copyright infringement by, among others, loading the source code of the CFI program onto a computer to inspect it.[1082] The court does not discuss why this act amounted to infringement. This aspect requires clarification.

[1079] *IBCOS Computers Ltd. and Another v Barclays Mercantile Highland Finance Ltd. and Others* [1994] F.S.R. 275 (*IBCOS*).

[1080] *Cantor Fitzgerald International v Tradition (UK) Ltd* [2000] R.P.C. 95 (*Cantor*).

[1081] See *IBCOS* at 302-3. See also *Cantor* at 131 [76], 133 and 135 [78]. A useful discussion of theses judgments is provided in Stokes S *Digital Copyright: Law and Practice* 3ed (2009) at para 6.3.2 and 6.3.3. See also the more detailed analysis in Caddick, et al. *Copinger & Scone James on Copyright* paras 7-83 to 7-92.

[1082] *Cantor* at 103 [10] and 104 [14].

The act of loading, which is usually associated with decompilation and one of the forms of incidental copying permitted by the exception, amounted to infringement in this case because it was not made by “a lawful user” of the program in terms of section 50B(1).[1083] In this case, Tradition did not have the right to use the code for any reason, including reverse engineering, because the copy was one retained by a former employee of CFI[1084] and, in addition, access to portions of the code was established by procuring a code from CFI by using his former username and password, without authorisation to do so.[1085] For this reason, CFI also raised two contentions of a breach of confidence,[1086] one of which argued that the use, without literal copying, of the code to correct errors in Tradition’s new program is unlawful.[1087] With this contention the court agreed,[1088] not because it amounted to copyright infringement (because it did not), but because the work used was obtained unlawfully and could, thus, not be used without breaching the confidence that was expected of a former employee.

These two findings of the court, regarding the lawful use of code for decompilation and the *use* of code for error correction, illustrate the interplay between contractual prohibitions and reverse engineering of computer programs. It places a clear emphasis on the fact that whatever exception is made to permit access to code, it must relate to a lawful copy and it must be used in a manner that does not violate any other confidentiality duty. These principles are inherent in the decompilation exception, as shown above, and emphasise the fact that the decompilation exception is primarily designed to protect the commercial interests of the copyright owner and not to facilitate access to unprotected elements or ideas. The modicum of access it does provide, is an absolute minimum required to suppress the justifiable needs of other programmers, but it does not balance the public and private interests.

[1083] At 139 [88].

[1084] At 101 [5] and 110 [36].

[1085] At 104 [15].

[1086] *Cantor* at 103 [12] regarding use of trade secrets and use of the code to correct the coding of Tradition’s program.

[1087] See Freedman C D “The Protection of Computer Software in Copyright and the Law of Confidence: Improper Decompilation and Employee-Poaching” 2000 *International Journal of Law and Information Technology* 8 (1) 2539-41 for a discussion of the confidentiality claims in this case.

[1088] *Cantor* at 138 [86].

This fact is made clear in *IBCOS* where the court held:

> "Source code, being what humans can understand, is very important to anyone who wants to copy a program with modifications, for instance to upgrade it. It is the source code which shows the human how it all works, and he or she will also get the benefit of all the comments laid down by the original programmer. Software houses not surprisingly **normally keep their source code to themselves and confidential**."[1089]

In this case, the court inferred that Mr Poole, the programmer who worked on both the original and the contentious programs, *had access* to the source code based on the common features and, in particular, the common errors in the original and the derivative programs, which indicated direct literal copying.[1090] Consequently, it held that the manner in which Mr Poole *used* the code, similar to the facts in *Cantor*, was not only an infringement of copyright but also "a breach of confidence [because] source code is normally kept confidential by software houses"[1091] and the proprietor (ADS) in this case considered the code a trade secret.

This aspect of these cases is seldom highlighted and not discussed in relation to the use of computer code after decompilation. This is understandable because both judgments are dominated by an analysis of the code or functions that were reproduced and the development of a discernment-methodology, a test for non-literal reproduction, to apply during the infringement test.

However, in the context of this work, the confidentiality aspect of these cases and its impact as a limiting factor on access to code by way of, or after, decompilation, is of crucial importance. As submitted above, it proves that the exceptions to copyright protection that facilitate access do not serve the public interest in gaining access. Instead, or in addition thereto, it has a distinct role in protecting the confidentiality interests of the software owner where copyright infringement does not protect its commercial interests. This is done by allowing the owner to argue, successfully, that it is a breach of confidence to *use*, by merely reading, the code.

[1089] *IBCOS* at 286 (emphasis added).

[1090] At 301.

[1091] At 314.

Of course, section 50B would apply to justify this use, but only under condition that the copy was *lawfully* used and only for the purpose of interoperability. This creates two problems for decompilation. First, it means that a program may not be decompiled and read if it was obtained in violation of a licence or other contractual limitation, as discussed above in relation to section 296 and the *Mars* case, even though the act of reading a work is otherwise clearly within the permitted acts. Second, it means that a program that is lawfully decompiled within the strict interoperability requirements may not be read and used to correct errors or other changes to a new program if the original program can be said to contain trade secrets. The interplay between trade secrets and copyright law is a discussion beyond the scope of this work. It suffices to note that the decompilation exception is not intended to be limited by the law on trade secrets and, clearly, did not consider its impact.

It is submitted that the breach of confidence argument, based on implied terms,[1092] is an imposition on the decompilation of computer programs and the monopolisation of programming techniques[1093] it effects is alien to copyright law principles, in particular the balance between private and public interest. As Freedman notes, it is particularly problematic in UK copyright law which "lacks the sophisticated structure developed in American law to assess claims based on reprehensible commercial conduct".[1094] This is even more true regarding South African copyright law.

4 4 Summation

This chapter has analysed, in detail, the two most prominent and developed decompilation exceptions to copyright infringement, established under two distinctly different regimes, based on fundamentally different doctrines of fair use and fair dealing and derived from two different sources of law, respectively case law and statute.

[1092] The court read the duty of confidentiality into the terms of the contract in *Cantor*. See Freedman *International Journal of Law and Information Technology* 40 for a discussion of the process applied by the court to classify the information regarding programming techniques as a trade secret.

[1093] See Freedman *International Journal of Law and Information Technology* 42.

[1094] 45.

It was important for this work to consider both because the prohibition on decompilation in South African law relies on the one system, namely, UK copyright law principles, while the proposed decompilation exception in South Africa will rely mainly on the other system, namely, US copyright law.

This chapter is not, however, a comparative legal analysis. The purpose is not to compare the two systems and arrive at a conclusion on which would be the most appropriate one to adopt in SA. On the contrary, this work seeks to formulate a new approach based on a contemporary, technically-sophisticated approach to decompilation that will also be manifestly more fair than either of the approaches outlined above.

Therefore, the research in this chapter sought to identify the common features of both systems, or the perspectives on computer programs that are present in both, which were responsible for the approach to decompilation adopted by both nations. By doing so, it was possible to identify a range of positive and negative features of the decompilation exceptions and the fairness considerations that play a part in its application. These factors will be useful later in this work to guide the construction of a better, fairer and more appropriate approach to decompilation.

The factors which negatively influenced the regulation of decompilation are:

- a fear that decompilation is an easy way to create reproductions of programs;
- an attempt to make the restricted act of translation applicable to computer programs because these works are expressed in linguistic characters;
- maintaining the widest possible range of acts which may be made subject to a copyright licence;
- safeguarding the economic interests of the copyright owner in relation to current and future works; and
- using copyright law to maintain a commercial advantage by locking out competitors.

On the other hand, this chapter also identified a number of factors that supported the need for a decompilation exception, all of which are useful when the fairness of an alternative approach to decompilation is considered. These findings are:

- it is difficult, if at all possible, to fully accommodate decompilation under copyright law if it is considered a type of literary work;
- the public interest in access to the ideas in code does not rely on the need to create derivative works, but on the need to read the work and understand how it functions;
- in all of the cases where decompilation occurred, and an infringing work was created thereafter, the decompilation exception had no impact on the rights of the copyright owner to prevent the reproduction of his code in another program;
- a prohibition on decompilation, or an overly-narrow decompilation exception, has the potential to extend patent-like protection by way of copyright law; and
- where the factors of the three-step test are applied, even if it is restricted to a statutory version of the test in national law, it consistently supports a decompilation exception as a necessity to maintain fairness.

In conclusion, while it is clear that decompilation is heavily regulated and dependent on continued extensive, technologically-sophisticated case law, it is apparent that neither the US nor the UK has managed to regard decompilation in a manner that is fair to both the copyright owner and the lawful user. This is largely due to the influence of the US in shaping international copyright law and the significantly outdated and limited understanding of computer programs upon which the principles of US copyright law regarding computer programs is based. The adoption of the US-inspired approach to a limited decompilation exception, albeit on a different basis, in the UK, and the similarities between these approaches and that of the TRIPS agreement and the WCT, are all due to the common view that computer programs are literary works.

Consequently, there is no evidence that the origins of the decompilation exception ever fully considered the *sui generis* nature of this form of exploitation. The volume of subsequent case law on decompilation, and the increasing complexity of the technical arguments made in these cases, is evidence of the fact that the decompilation exception is unsophisticated and insensitive to the reality of computer programs. The additional layer of complexity, which is added by the imposition of an anti-circumvention provision, and the rise of a substantial argument based on copyright misuse or abuse as a defence to infringement in computer programs, is further proof that the approaches to decompilation identified in this chapter are ill-suited to copyright law principles, consistently fails to achieve a fair balance between the public and private interests and routinely violate the idea/expression separation.

Chapter 5

Developing a Fair Decompilation Exception in South African Copyright Law with Reference to the Three-step Test

5 1 Introduction

In the preceding chapters it has been shown that the act of decompilation does not technically amount to the making of an adaptation or a translation and, therefore, does not fit within the framework of copyright protection for this type of work as long as it relies on a literary-works analogy. A deep analysis of foreign law has shown that, in order to sustain the protection of computer programs as literary works and permit access to the source code by decompilation, it must rely on a complex system of judicial balancing either under a general fair use regime or within the confines of a fair dealing statutory exception. The legal analysis exposed a marked disparity between the two approaches and laid bare the plethora of problems that the need for decompilation still poses to copyright law, most of which remain unsolved. This makes it clear that the act of decompilation is, despite wide and deep judicial consideration, still difficult to accommodate within the copyright regime. This is made significantly more difficult where the protection of the work relies on a literary-works analogy.

Thus, despite the efforts made above to use the *sui generis* classification of software as a basis to develop a decompilation justification, and the evidence provided above that non-literal elements remain protected despite the introduction of a decompilation exception, a last question remains to be answered in order to address the topic of this work.

The *sui generis* classification of computer programs in South African copyright law makes it possible for the law to develop individually from foreign law, while remaining within both settled copyright law principles and international consensus. And yet, as has been shown, there is a pressing need for copyright law in South Africa to become more flexible and attuned to national interests. This is the *impact* of copyrightable non-literal elements that are at issue in this work – where the law operates, directly, by prohibiting decompilation, or indirectly, by limiting access or use, the idea/expression dichotomy is tilted in favour of the rightsholder and imperils the public interest.[1095] This risk is the main argument addressed in this chapter. It is proposed that the idea/expression separation implies a fairness test for all exceptions which must meet the standards of, and exploit the flexibilities inherent in, the three-step test.

5 2 The fairness analysis

The idea/expression dichotomy is a consequence of the nature of copyright and the inherent restriction to expressions. It is, also, a summation of the intrinsic value test that justifies copyright protection. Therefore, a consideration of the idea/expression dichotomy in relation to any work or case of infringement must, and will inevitably, focus on the question of what is *fair* in the particular case. In this context, fairness refers to the balance between, on the one hand, affording suitable protection for the results of individual labour and the continued safeguard thereof against an erosion of the statutory right and, on the other hand, leaving the inspiration and creative thought that inspired the work, or might inspire others, both accessible and free to use.

For this reason, the idea/expression dichotomy is at play every time a copyright is exercised or infringed and, in each instance, where the existence or scope of the right is challenged by the public interest or developments in technology. Thus, even where a copyright regime, such as, that of the UK, resists a formal recognition of the dichotomy, it is nonetheless present in legal development.

This is because the idea/expression dichotomy is built into the primer for all copyright law, namely, the Berne Convention, in the form of the three-step test.[1096] This test

---

[1095] In respect of the specific public interest, namely access to knowledge for a range of social and cultural reasons, that underpins the analysis in this chapter, see the discussion of the ESC model in chapter 2 above and, in particular, the discussion at fn 22 and 24.

[1096] Berne Convention Article 9(2):

> "It shall be a matter for legislation in the countries of the Union to permit the reproduction of such works in certain special cases, provided that such reproduction does not conflict with a normal exploitation of the work and does not unreasonably prejudice the legitimate interests of the author."

embodies the fairness balance and is, as such, the closest the law can get to a clear demarcation between idea and expression – it is *the* test for fairness.[1097]

Therefore, in order to address the idea/expression dichotomy in this work, the three-step test is the final measure of accuracy and legality for the findings made in this work. It is not sufficient to submit, as this work has done, that decompilation may be technically and legally warranted under the *sui generis* system in South Africa or that the models for a decompilation exception borrowed from foreign law are ill suited to South Africa – it must also be proven that the idea/expression dichotomy support these findings.

In addition, the importance of the three-step test is specifically stressed as the balancing mechanism for copyright works in relation to digital work and infringement cases[1098] as *the* means "to devise new exceptions and limitations that are appropriate in the digital network environment."[1099] The three-step test is also recognised as the only way to incorporate a fairness analysis in copyright law that is acceptable to both fair use and fair dealing regimes precisely because of the flexibilities established by the wide wording of the test.[1100]

Furthermore, the three-step test is made expressly applicable to decompilation. Article 6(3) of both Software Directives[1101] states that where a decompilation exception is created in national law:

[1097] Geiger C et al "The Three-Step-Test Revisited: How to Use the Test's Flexibility in National Copyright Law" 2004 *American University International Law Review* 29 (3) 581 at 592 where the authors point out that, following the incorporation of the three-step test in TRIPS, and subsequent review of national law by the WTO "the three-step test became one of the main, if not *the main issue*, when trying to find a fair balance of interest in copyright law and policy."

[1098] Agreed Statements Concerning the WIPO Copyright Treaty adopted by the Diplomatic Conference on 20 December 1996: Concerning Article 10.

[1099] Agreed Statement Concerning Article 10.

[1100] Geiger *American University International Law Review* 591.

[1101] Council Directive on The Legal Protection of Computer Programs; Directive of the European Parliament and of the Council on the Legal Protection of Computer Programs.

> "The provisions of this Article may not be interpreted in such a way as to allow its application to be used in a manner which unreasonably prejudices the rightholder's legitimate interests or conflicts with a normal exploitation of the computer program."

5 2 1 The status of the three-step test

The origin of the three-step test in article 9(2) of the Berne Convention is said to have originated as "a counterweight to the formal recognition of a general right of reproduction at the 1967 Stockholm Revision Conference"[1102] in order to regulate the making of exceptions and limitations.

The open-ended wording of the test is the result of a compromise to accommodate both civil-law countries that preferred a closed list of statutory exceptions and other countries that favoured a more flexible and general test. Curiously, despite its open-ended wording, the text of the test is based on a formulation submitted by the UK in English.[1103]

The test has since been adopted, largely unamended, in, among others, article 13 of the TRIPS agreement and article 10 of the WCT, which means it must be read within the context of the purpose and spirit of all three documents and interpreted consistent with the other provisions of both TRIPS and the WCT. Article 13 of TRIPS has also expanded the application of the test beyond the right to reproduction and makes it applicable to exceptions or limitations of all rights[1104] including both the rights granted under the Berne Convention and any new rights established in terms of TRIPS.[1105]

[1102] Geiger *American University International Law Review* 583.

[1103] 584.

[1104] WTO Dispute Resolution Panel (2000) *Report of the Panel on United States – Section 110(5) of the US Copyright Act* 27 [6.74] (WTO *Panel Report* or WTO Report).

[1105] WTO *Panel Report* at 28 [6.80]. The three-step test in the Berne Convention does not apply to rights granted in terms of articles 2(4), 2(7), 2(8), 2*bis*, 10, 10*bis* and 13(1). However, the three-step test as incorporated in TRIPS, will apply to rights granted under these terms, depending on the facts of the case. See WTO *Panel Report* 30 [6.91]. None of the rights to which the three-step does not, or may not, apply, are relevant to this work.

Because South Africa is bound by the Constitution[1106] to give effect to the Berne Convention and the TRIPS agreement, and soon to be bound to the WCT, the interpretation of the three-step test in relation to these documents is an imperative in considerations of statutory exceptions. The manner in which international law is read and applied is made clear in the Vienna Convention[1107] to the extent that the treaties must be read as a whole and "in their context and in the light of its object and purpose"[1108] and including its preamble and annexures.[1109]

5 2 2 The nature of the three-step test

As its common name denotes, the test consists of three elements that must be considered when an exception or limitation to copyright is considered. The measure must: (1) relate to a certain *special case*; (2) not conflict with the *normal exploitation* of the work; and (3) not *unreasonably prejudice* the legitimate interests of the author.

This wording has, for a long time, remained vague and intentionally open-ended, although it has been applied in great detail in many cases.[1110] However, in 2000, a dispute resolution panel of the World Trade Organisation delivered its report on the exceptions in section 110(5) of the US Copyright Act,[1111] which made it clear that the three-step test is not only binding on all Berne, TRIPS and WCT members but that non-adherence to the test will result in nullification of a provision in national law.

The WTO Report provides, for the first time, a detailed analysis of the purpose, nature and content of the three-step test. The findings of the report, in relation to each step,

---

[1106] The Constitution of the Republic of South Africa 1996 Sections 39(1)(b) and 231(4) and (5).

[1107] Vienna Convention on The Law of Treaties No 18232 (23 May 1969).

[1108] Article 31(1) of the Vienna Convention. See also WIPO *Panel Report* 17 [6.43].

[1109] Article 31(2).

[1110] See for example the discussion of the judgment in *Sega* and *Sony* above in paragraphs 4 3 2 9 and 4 2 3 10 respectively, where the features of the three-step test were used to guide the interpretation of the fair use test in US law. It has also been central to the development of the abstraction/filtration tests in both US and UK case law. See for example the decisions in the *Whelan* and *Atari* cases, discussed above in chapter 4.

[1111] WTO Dispute Resolution Panel (2000) *Report of the Panel on United States – Section 110(5) of the US Copyright Act*.

are summarised and discussed below. At the outset, it should be noted that the panel delivered substantive commentary on the legal interpretation of the test independently of the facts in dispute. Therefore, the WTO Report, insofar as it deals with the interpretation of the test itself, contains objective findings on the meaning of each step of the test. For this reason, it is not appropriate or necessary to deal with the facts of the dispute that initiated the report.[1112]

5 2 2 1 Certain special cases

Although it has been generally accepted that this requirement means that the exception must be clearly defined in statute to the extent that it is legally certain[1113] because "the scope of the exception is known and particularised,"[1114] the WTO Report made it clear that this alone is not sufficient. In order to qualify as a legitimate exception under the test, the case must be both certain *and special*.

In this respect, a limitation will only be special if it is "**narrow** in a quantitative as well as a qualitative sense,"[1115] which means it must have a *particular objective* in mind. The qualitative factor places an emphasis on the *purpose* of the special provision. Although this does not imply that the first step includes a value judgment on the legitimacy of the exception under consideration, it means that "public policy purposes stated by law-makers when enacting a limitation or exception"[1116] play a role in determining whether the exception serves a special purpose. Where the exception relates specifically to the reproduction right, the WTO Report recognises that the *public policy purpose* of the exception is of particular interpretive value.[1117] In other words, an exception that displays a "valid public policy or other exceptional circumstance that

[1112] For a comprehensive analysis and opinion on the panel's findings in relation to the facts of the dispute, see Ginsburg J C "Toward Supranational Copyright Law? The WTO Panel Decision and the 'Three-Step Test' for Copyright Exceptions" 2001 *Revue Internationale du Droit d'Auteur* 187 3.

[1113] Geiger *American University International Law Review* 593.

[1114] WTO *Panel Report* 33 [6.108].

[1115] 33 [6.109] (emphasis added).

[1116] 34 [6.112].

[1117] 33 [6.111].

makes it inappropriate or impossible to enforce the exclusive rights conferred"[1118] would be more likely to be considered sufficiently special.

Regarding the quantitative factor, a special exception must be limited with respect to "the scope in respect of potential users,"[1119] where users refer to the parties who will benefit from the exception. On the facts of this case, the panel found that this factor is not met where "a substantial majority"[1120] of users, who are otherwise prevented from carrying out the restricted act, will benefit from the restriction. In other words, the exception is not warranted, or special, if it effectively nullifies the restricted act. This question goes to the heart of the restricted act and its purpose. Where the restricted act is affected by the exception to the extent that it is rendered ineffective in the *substantial majority* of cases, the exception fails to pass the first step. For example, in this case where the exemptions permitted the transmission or public performance of music in eating and drinking establishments and retail outlets, it was shown that it was applicable to all such establishments regardless of its size.[1121] This, the panel found, was not quantitatively narrow enough.[1122]

5 2 2 2 Not conflict with a normal exploitation of the work

This criterion is traditionally applied as the lead factor in limiting the availability, and scope, of exceptions to restricted acts. It is applied as both an internal and external limitation by determining whether or not an exception shall be available and, if so, the extent to which it shall apply.[1123] As such, it is the basis of inherent limitations in exceptions, such as, the interoperability criteria in the decompilation exception.

[1118] 42 [6.154].

[1119] WTO *Panel Report* 37 [6.127].

[1120] 38 [6.133].

[1121] 38 [6.133]. See 3 [2.3] and 4 [2.5] of the WTO report for the original wording of the exception in section 110(5), which "purpose is to exempt from copyright liability anyone who merely turns on, in a public place, an ordinary radio or television receiving apparatus of a kind commonly sold to members of the public for private use."

[1122] WTO *Panel Report* 37 [6.129] and 38 [6.133]. Despite failing at the first step, the panel nevertheless considered the exception in detail regarding the other two factors, so that its recommendations may be used as an interpretive measure in giving future rulings. See WTO *Panel Report* 43 [6.161].

[1123] WTO *Panel Report* 45 [6.170] to [6.171].

The original intention of this factor was, according to the WTO Report, to protect the ability of creators "to extract economic value from their rights to those works"[1124] and that the word "works" must be read to mean "exclusive rights."[1125] Thus, this factor considers two things. First, whether the exception has a detrimental impact on "the forms of exploitation that *currently generate income* for the right holder as well as those which, in all probability, *are likely to be of considerable importance in the future*."[1126] Second, the factor considers whether this impact relates only to one of the exclusive rights or more than one.[1127]

However, in light of the objectives of TRIPS, the WTO Report makes it clear that the "'normal' exploitation clearly means **something less than full use** of an exclusive right"[1128] and, instead, refers to the reasonable manner in which the owner can expect to derive an income from restricting the use of the work.[1129]

In relation to exceptions that seek to limit the 'normal exploitation' of a work, that is to say where the exception imposes a limitation on a branch of direct income from the use of the work, TRIPS provides that the exception may be warranted and, therefore, pass the second factor of the test, where it has a developmental objective which serves a national policy interest. In other words, this factor is not to be applied, as it has frequently been done, as merely "an economic analysis of the degree of 'market displacement' in terms of foregone collection of remuneration by right owners caused by the free use of works due to the exemption at issue."[1130] According to the WTO Report, the weighing of this factor requires "a more normative approach to defining normal exploitation, that includes, inter alia, a dynamic element capable of taking into account **technological and market developments**."[1131]

[1124] 44 [6.165].

[1125] WTO *Panel Report* 45 [6.171].

[1126] Geiger *American University International Law Review* 594 (original emphasis).

[1127] WTO *Panel Report* 45 [6.170].

[1128] 44 [6.167].

[1129] 45 [6.168].

[1130] 47 [6.177].

[1131] 47 [6.178] (emphasis added).

In other words, when applying the second factor and determining whether the anticipated exception will impact the normal or reasonable exploitation of the work, it may not be argued that the exception must *per se* fail because it can be shown that it will limit the economic interests of the owner. This is because the three-step test must be read consistently with the objects of TRIPS and the WCT. Thus, where either of these seek to promote an objective that will, or may, justify an imposition on a particular revenue stream from a work, the three-step test must be read consistently with such objective and may not present a barrier to the making of an exception that seeks to pursue these objectives. This applies to the normal exploitation at the time the exception is made *and* the possible future sources or means of deriving an income from the work that may arise.

In the context of this work, the key objectives in TRIPS, which must be read as factors that may justify a limitation on the normal exploitation of the work, are: (1) "the underlying public policy objectives of national systems for the protection of intellectual property, including developmental and technological objectives;" and, (2) "maximum flexibility in the domestic implementation of laws and regulations in order to enable them to create a sound and viable technological base".[1132] The South African perspective on developmental and technological objectives that affect law-making on copyright issues have been identified and discussed in this work above.[1133]

Thus, where the developmental objective and policy interest relate to technological advances, the clear intention of TRIPS and the WCT is to "shelter a number of key use privileges"[1134] among which are the accessibility of the work as a building block for future work and the advancement of a sphere of economic activity. Insofar as the interpretation of these use privileges are reflected in South African policy[1135] on copyright exceptions, they must be read in light of the flexibilities afforded by an interpretation of international law on this factor of the test.

1132 TRIPS Preamble.

1133 See paragraph 2 1 3 above.

1134 Geiger *American University International Law Review* 590, with reference to the Minutes of Main Committee I during deliberation on the exceptions to the WCT and WPPT, which, the author correctly submits, "mirror" the ideal of TRIPS in this respect.

1135 See the discussion of South African policy at paragraph 2 1 2 above.

However, the WTO Report maintains that the second factor remains an essentially economic consideration. In light of the "normative standard" which TRIPS and WCT made applicable to this factor, it considers "those forms of exploitation that currently generate **significant or tangible revenue** [and] those forms of exploitation which, with **a certain degree of likelihood and plausibility**, could acquire **considerable** economic or practical importance."[1136]

In other words, where an exception seeks to promote a developmental goal or other policy objective, it is more likely to pass the second step of the test if it relates only to a secondary or less significant revenue stream at that moment, or, to a revenue stream that is not likely to become important.

For example, an open-access exception to permit the wholesale reproduction of textbooks for educational use (a reading exemption) would not pass the test because it relates to the primary revenue stream of the work. In this case the developmental goal or policy objective cannot outweigh the conflict it causes with the normal exploitation. Conversely, the same open-access exception to permit temporary reproduction of code in order to read it, would pass the test because it relates to a secondary and less significant revenue stream derived from control over providing accessible format copies of the work.

In addition, such an exception will be "presumed not to conflict with a normal exploitation of works if they are confined to a scope or degree that does not enter into economic competition with non-exempted uses."[1137]

Once again, a reading exemption fails in relation to textbooks because it annihilates the commercial value of the work while the same exemption in relation to computer programs survives the test because it does not create an avenue for competition or restrict the commercial value of the work.

[1136] WTO *Panel Report* 48 [6.180].

[1137] 48 [6.181].

The WTO Report makes it clear that the competition element does not mean that "every use of a work, which in principle is covered by the scope of exclusive rights and involves commercial gain, necessarily conflicts with a normal exploitation of that work"[1138] and that the consideration of the commercial element may result in an exception being justified in one country or industry but not in another. This is because the "'normalcy' of a form of exploitation should be analysed primarily by reference to the market of the WTO Member whose measure is in dispute".[1139]

Therefore, the inherent flexibility of the three-step test must be safeguarded against an overriding commercial consideration, otherwise the test would serve no function.[1140]

5 2 2 3 Not unreasonably prejudice the legitimate interests of the right holder

In this context, the third factor guides the interpretation of the commercial considerations identified in factor two and "offers considerable flexibility for the balancing of competing interests."[1141]

According to the panel, this factor measures whether the prejudice, which is theoretically present in all exceptions, is *unreasonable*[1142] regarding only those interests that are *legitimate*, as opposed to justified[1143] "on the basis of the **economic effects in the country** applying the exception."[1144]

Therefore, the application of this factor requires an identification of which interests are affected *and* the "attributes" which make them legitimate.[1145] These questions must be

---

1138 WTO *Panel Report* 48 [6.182].

1139 50 [6.189].

1140 WTO *Panel Report* 48 [6.182]. If this is allowed to happen, "article 13 might be left devoid of meaning, because normal exploitation would be equated with full use of exclusive rights."

1141 Geiger *American University International Law Review* 595.

1142 WTO *Panel Report* 57 [6.220].

1143 57 [6.221].

1144 57 [6.221] (emphasis added).

1145 57 [6.222].

answered at a national level, in light of the flexibility afforded by the test itself and by other provisions of the treaties which contain, and provide context for, the test.

This means that the third step is "a refined proportionality test"[1146] which "relates to lawfulness from a legal positivist perspective [and] legitimacy from a more normative perspective."[1147] In other words, the legitimate interests must be weighed not only in relation to its statutory basis but also the "objectives that underlie the protection of exclusive rights." In this context, the crucial objective at issue in this work is the appropriate balance of public and private interests, manifested in a wider accessibility of knowledge and the teaching contained in source code.

Regarding how the *degree* to which the prejudice, caused by the exception, should be measured, the panel does not propose a methodology of identifying the line beyond which a limitation on the right becomes unreasonable. The WTO Report specifically applies this factor as an analysis of "market conditions",[1148] and not market impact. This makes it clear that the omission of a hard line between reasonable and unreasonable was done deliberately in order to promote the flexibility of the test, which allows each member state to make its own determinations in light of its national policy considerations.

Furthermore, the WTO Report states that the use of "unreasonable" indicates a higher, or stricter, standard of prejudice than not reasonable.[1149] Considering that the latter term means something that is not fair, the panel's submission implies that the use of the term "unreasonable" indicates that an exception should not only be measured, under the third factor, in light of its impact on absolute fairness (between user and owner interests) in a legal positivistic sense, but also in light of what is fair in a normative sense. In other words, the flexibility inherent in this factor means it does not seek to protect the economic interests of the owner because that is legally fair, or

---

[1146] Geiger *American University International Law Review* 595 defines this test as follows:

> "The legitimacy of the interests invoked by authors and right holders are to be weighed against the reasons justifying the use privilege."

[1147] WTO *Panel Report* 58 [6.224].

[1148] 61 [6.236].

[1149] 58 [6.225].

justified by the law, but because it has an effect that is fair. This effect is measured in light of national conditions. This does not mean that fairness may be measured in light of external factors only – the fairness which supports an exemption in relation to the interests of users must also be fair toward the owner.

Thus, for example, it is inappropriate to suggest that a reading exception is fair because it seeks to alleviate inequality and promote education where a historical disadvantage exists. This would focus only on the external fairness factors. It must also be shown that this external fairness factor will be fair to the copyright owner, despite the imposition on his/her rights. So, conversely, a reading exception *for computer programs* will be fair because it addresses the need to have access, in a developing economy, to sophisticated programming methods and ideas in order to boost a local industry. This is an external fairness factor which is also fair to the copyright owner because it accords with the letter and spirit of copyright law to promote a balance of interests. By making the reading exception specific to a type of work, it gives effect to both internal and external fairness factors and the developmental goals (industry stimulation) and policy considerations (education and alleviation of poverty or inequality). To include an additional limitation in the exception, such as interoperability, would be unfair, in light of the flexibility imperative of the test, because it allows the internal fairness considerations (of the owner) to outweigh external fairness considerations.

That is not to say that the internal and external fairness considerations must always be equally matched. This would be contrary to the meaning of the test as a whole which seeks a balance, not an equilibrium. It would also be contrary to the WTO panel's finding that the developmental objects of TRIPS and the WCT, which serve to promote external fairness considerations, are to be read into the test throughout. All it means is that the combination of external and internal fairness considerations must be *related.* No one side may be allowed to completely outweigh the other and both must consider the essential and inherent balance between proprietary interests and the increase in knowledge that copyright seeks to maintain.

Because the wording implies a wider analysis of fairness, it means that the third factor will only delegitimise an exception where it is shown to be unfair to the owner and failing to promote a balance of the public interest. In other words, the prejudice caused by the exception must be fair to both the owner and the users of the work. Put differently, the prejudice would only be unreasonable if it is not a fair way of balancing the private and public interest. It is, therefore, not just a question of whether the exception is unreasonable to the owner – this is only a part of the bigger reasonableness and proportionality question posed by the third factor.

5 2 3 The application of the test

The composite parts of the test are to be read "on a cumulative basis"[1150] and interpreted as conditions,[1151] all of which must be met in order to justify a limitation in terms of article 9(2) of Berne and article 13 of TRIPS.

The three parts of the test were intended, by the drafters of the Berne Convention, to apply as consecutive steps, each of which may be a disqualifying criterion. Therefore, a limitation or exception must meet all three criteria, but will fail if it does not meet any one of the criteria.[1152] Consequently, there is no possibility to argue that failure on any one step is corrected by meeting or exceeding the conditions in another.

Strong academic opinion exists[1153] that, particularly after the introduction of TRIPS and in order to "give judges sufficient latitude for considering other interests than the right holders',"[1154] the test should be read as three interrelated and interchangeable *factors,* rather than steps or stages.[1155] In light of the extensive normative evaluations

---

[1150] WTO *Panel Report* 27 [6.74] and 31[6.97].

[1151] 27 [6.74].

[1152] Main Committee I of the Stockholm Diplomatic Conference (1967) as cited by the WTO *Panel Report* at 27 [6.73].

[1153] Senftleben M "The International Three-Step Test: A Model Provision for EC Fair Use Legislation" 2010 *Journal of Intellectual Property, Information Technology and Electronic Commerce Law* 1 67 73 para 39; Griffiths J "The "Three-Step Test" in European Copyright Law – Problems and Solutions" 2009 *Queen Mary University of London, School of Law Legal Studies Research Paper No 31/2009* 10; Geiger *American University International Law Review* 606.

[1154] Koelman K J "Fixing the three step test" 2006 *European Intellectual Property Review* 28 (8) 407 408.

[1155] The strongest opinion, repeated by others listed above, is that of Koelman *EIPR* at 410:

added to the test by the WTO Report, there is some logical sense in these arguments for a more flexible, factor-based approached to the test. However, the panel did not endorse this view. On the contrary, in its findings on the facts, the panel made it clear that the exception failed on the first step and, therefore, no further analysis of the particular exception in light of the next two steps was required.[1156]

In addition, the WTO panel stated that the "principle of effective treaty interpretation requires us to give a distinct meaning to each of the three conditions and to avoid a reading that could reduce any of the conditions to redundancy or inutility."[1157] It is submitted that the approach of the panel remains correct. It has provided ample room for flexibility within each step. For example, the opinion of scholars that the second step, the normal exploitation criteria, should be read narrowly, "to avoid depriving national legislatures and judges of the policy space necessary to strike a proper balance between copyright protection and competing social, cultural, and economic needs"[1158] has been clearly incorporated by the panel.

There is no clear reason why the structure of the test should be altered from a stepped approach to a factored approach in order to provide more flexibility. Creating the possibility that one step may override another will result in arbitrary results and negate the function of the test as a coherent and consistent fairness test. Beyond this point, the current work does not require an analysis of possible alternative interpretations of the test. It will suffice to take the opinion of the WTO panel as dispositive.

5 3 The future of decompilation in South African law

The first part of the fairness analysis considers whether, in light of the findings above, the proposals for law reform in relation to decompilation adequately address the tension expressed by the idea/expression dichotomy. Regardless of whether or not

> "The solution thus becomes obvious: the three-step test should be redesigned in a way that converts the three hurdles to factors that courts or legislators have to weigh together when deciding on the scope of copyright."

[1156] WTO *Panel Report* 43 [6.160].

[1157] WTO *Panel Report* 31 [6.97].

[1158] Geiger *American University International Law Review* 603 fn83 citing Senftleben M *Copyright, Limitations and the Three-Step Test* (2004) at 193.

these particular reforms are enacted, the observations made below will remain relevant to any future legal development because they consider the principles of the law rather than the precise wording of the law. Therefore, the proposed amendments serve only as an example, or case study, of how decompilation should, or should not be, accommodated in copyright law.

5 3 1 The statutory decompilation exception

The 2017 Copyright Amendment Bill[1159] will introduce a statutory exception for decompilation based on a mixture of the wording of the Copyright Directive, the CDPA and the WCT. It will repeal all of the current fair dealing exceptions in relation to computer programs and replace section 19B of the principal act with the following provisions. Section 19B(1) introduces a general reverse-engineering exception by observing the operation of the program. Section 19B(2) introduces a statutory decompilation exception. The latter is the focus of this discussion.

> "19B. (1) A person having a right to use a copy of a computer program may, without the authorization of the copyright owner, **observe, study or test the functioning of the program** in order to determine the ideas and principles, which underlie any element of the program if that person does so while performing any of the acts of loading, displaying, executing, transmitting or storing the program, which he or she is entitled to perform.
>
> (2) The authorization of the copyright owner shall not be required where **reproduction** of the code **and translation** of its form are **indispensable** in order to obtain the information necessary to achieve the **interoperability** of an independently created computer program with other programs, if the following conditions are met:
>
> > (a) The acts referred to in subsection (1) are performed by the licensee or another person **having a right to use** a copy of the program, or on their behalf by a person authorized to do so;
> >
> > (b) the information necessary to achieve interoperability has not previously been readily available to the persons referred to in paragraph (a); and
> >
> > (c) those acts **are confined to the parts of the original program which are necessary in order to achieve interoperability**."[1160]

[1159] Copyright Amendment Bill [B 13B—2017] (November 2018 version) (2017 Copyright Amendment Bill).

[1160] Section 19 of the 2017 Copyright Amendment Bill.

5 3 1 1 A special case

As a general proposition, it has been well established in case law in the US and UK that a decompilation exception is a special and narrow case on, at least, the bases that: (1) it relates to an activity that is peculiar to this type of work, (2) it is necessitated in limited circumstances where the form of the work is not humanly legible and (3) it has a specific purpose, namely, to provide access. Since section 19B(2) mirrors the wording of these provisions, it can be assumed with certainty that it passes the first step. This does not mean that the exception is fair in relation to the first step.

As was shown above, this specific purpose has been further narrowed to a single instance of access for the sub-purpose of creating an independent and interoperable work. The decompilation exception in section 19B(2) will clearly pass the first step of the test, insofar as it is narrow and specific.

However, the copying of the additional narrowing factor, the interoperability limitation, from foreign law is unwelcome and unfair, because it fails to give effect to the flexibility proposed by the three-step test. This point will be discussed further in relation to the second and third steps. However, it should be noted that the first step requires only that the exception be narrow and special in relation to its purpose for the sake of legal certainty. Thus, the first step does not suggest that the purpose of the exception must itself be narrowed.

5 3 1 2 Not conflict with a normal exploitation of the work

The exception applies to two exclusive acts, namely, reproduction and adaptation. The potential for conflict is thus, in principle, doubled.

As shown above, the limitation on reproduction does not in any way relate to the “normal” exploitation associated with copying, because decompilation involves only *intermediate* or transient reproduction of the object code, while the source code is not copied at all.

The potential for a conflict with the normal exploitation may occur where a derivative work is made after the process of decompilation has occurred, by literal text borrowing. However, this secondary, tangential potential conflict is adequately addressed by the test for infringement, and there is no clear reason advanced in case law or jurisprudence why the decompilation exception will result in a substantial conflict on this basis.

Furthermore, the technical analysis of decompilation above has proven that it does not involve the making of a translation of the object code, which constitutes an adaptation of the source code. The process delivers, at best, an approximation of the source code. Thus, there may only be said to be a conflict with the normal adaptation right if the existence of the decompiled source code somehow imposes on the right of the copyright owner to make adaptations of the code in a manner that he/she would *normally* do in order to extract value from the work.

Clearly, it cannot be said that the act of making a simulacrum of the source code is within the *normal* range of activities that a copyright owner carries out on a commercial basis. The adaptation right of the copyright owner relates only to making direct translations of its own source code into object code for the purpose of creating functional programs. Nothing in this right relates to making educated guesses about other possible adaptations which are, per se, non-functioning, as the basis of the normal economic activity of the owner.

Thus, the exception clearly passes the “normal exploitation” element of the second step. However, as shown above, this step is no longer a purely economic consideration and has a particular normative character which prohibits the application of this step as an overriding commercial or competitive protection measure.

As a fairness measure, therefore, section 19B(2) fails this step of the test because it does not illustrate that the developmental objectives of South Africa are being served by the exception. The fact that the wording of the exception has been copied, make it clear that this has not been done.

It has been shown above that the developmental objectives of South Africa in relation to computer programs are, firstly, access to knowledge for learning and, secondly, the stimulation of a local software producing industry. These objectives differ significantly from those that underlie the wording of this provision, which are the risk of market displacement and/or competition.

Unlike the UK and US, where this wording was developed, South Africa is a net-consumer of software. Thus, the imperative to safeguard local software owners against derivative works that may compete in the same market segment is not present, at least not nearly to the extent it is present in the US or UK.

Furthermore, the need to protect foreign software owners against market displacement or competition is highly unlikely in a developing nation which does not possess the ability to make sophisticated software that could displace currently licensed work. The likelihood that this may occur in future is remote, and there is no reason to suggest that the policy objective of South Africa can be served while the interoperability limitation exists.

Removing it from the exception will not create an avenue for piracy because the test for infringement remains unaffected and the scope of protection afforded to computer programs has not been eroded. Foreign software owners, therefore, remain in as strong a position in relation to their work in South Africa as before, even if the interoperability limitation is removed. It will also allow wider compatibility of locally produced new work and force licensing costs of foreign work down to a more competitive level.

The WTO panel made it clear that the limitation on the normal exploitation of the work must be decided in light of *local,* national market factors. There is no indication that this has been done. On the contrary, by retaining the interoperability limitation, there is strong reason to submit that the exception is unfair in light of the second step of the test, because it protects an interest of the copyright owner that is neither normal nor in conflict with the activity that the exception seeks to legitimise and does not give effect to the normative considerations inherent in this step.

5 3 1 3 Not unreasonably prejudice the legitimate interests of the right holder

Section 19B(2) also fails the fairness analysis on the third factor. The WTO panel made it clear that this is a proportionality test regarding the legitimacy of the interest being protected and the reasonableness of the limitation on that interest, both of which must be measured in order to establish a fair balance between the competing interests in the work.

Regarding the legitimate interest in this case, it is clear that the exception is not fair because it serves only the interest of the rights holder in a developed economy perspective. This is contrary to the WTO report, which stated that the legitimacy of the interest must be measured in light of the objective that underlies the relevant exclusive right. In this case, the objective of protecting computer programs against reproduction is purely economic. By imposing a limitation on the *use* of that work, by decompilation for only a narrow purpose, the economic objective of the owner is served at the expense of those who wish to, or need to, decompile the program for a purpose other than interoperability. This is in conflict with the objective of copyright in South Africa which, as shown above, also consider the protection of work important from a social perspective. Where a limitation is too narrow in a particular case, the exception fails to be fair.

In addition, the economic objective of copyright in computer programs in a global or foreign perspective may not be allowed to outweigh this factor because the panel made it clear that the legitimacy of the limitation must be measured in light of the economic effect it will or may have in the country where it is implemented. Once again, the manifestly different local software market, compared to those where this limitation originated, make it clear that the interoperability limitation is unjustified.

In this context, the limitation must also be examined for reasonableness. The panel submit that it is the market conditions in the country, not the market impact of the exception, that must guide the drafting of the limitation. In this context, it must show a clear balance between the internal and external fairness factors. The internal fairness factors in the case of decompilation, suggest that interoperability is necessary to prevent any abuse of the decompiled code. As shown in detail above, this is neither likely nor justified by copyright law principles. The external fairness factors indicate that decompilation should not be limited by interoperability because it transgresses the idea/expression dichotomy and prevents all but one form of fair engagement with the work.

Thus, there is no question that the exception is unfair. While the interoperability limitation guarantees that the exception cannot be said to impose an unreasonable limitation, its presence makes the exception unfairly narrow. As a result, there is clear evidence that a decompilation exception without such a limitation will continue to be acceptable under the third step of the test and will not impose an unreasonable limitation on the rights holder.

5 4 Fairness analysis conclusion

It is clear that, while section 19B(2) may have passed the three-step test in other jurisdictions, it fails to give effect to, or use, the inherent flexibilities of the test in the South African context. Consequently, it does not meet the standard of fairness regarding the second or third factors and places an undue restriction on the use of computer programs. This is, it is submitted, an abuse of copyright law and a missed opportunity to give effect to national development goals without derogating from the full enjoyment of copyright protection.

5 4 1 The fair use decompilation exception

The 2017 Copyright Amendment Bill will, in addition to the fair dealing exceptions in the current act and those introduced by the Bill, among which is section 19B, also introduce a fair use regime by means of the proposed section 12A.[1161]

[1161] 2017 Copyright Amendment Bill:

"12A. (a) In addition to uses specifically authorized, fair use in respect of a work or the performance of that work, for purposes such as the following, does not infringe copyright in that work:

> (i) Research, private study or personal use, including the use of a lawful copy of the work at a different time or with a different device;
>
> (ii) criticism or review of that work or of another work;
>
> (iii) reporting current events;
>
> (iv) scholarship, teaching and education;
>
> (v) comment, illustration, parody, satire, caricature, cartoon, tribute, homage or pastiche;
>
> (vi) preservation of and access to the collections of libraries, archives and museums; and
>
> (vii) ensuring proper performance of public administration.

(b) In determining whether an act done in relation to a work constitutes fair use, all relevant factors shall be taken into account, including but not limited to—

> (i) the nature of the work in question;
>
> (ii) the amount and substantiality of the part of the work affected by the act in relation to the whole of the work;
>
> (iii) the purpose and character of the use, including whether—

This creates the possibility that a South African court may develop a decompilation exception in a manner similar to US case law on this point. In particular, the wording of section 12A(b)(iii)(aa), whether the fair use results in a work which "serves a purpose different from that of the work affected", could form the basis of a transformative-use exception.

In the, unlikely, event that such a case is made, the court would be required to balance the exception with the provision in section 19B(2). If the latter provision retains the interoperability limitation, the transformative-use exception may permit decompilation for another purpose.

Considering that no such exception exists at this time, it is impossible to apply the three-step test to section 12A as a fairness measure, unless it is done hypothetically, which this work will not do.

A detailed analysis of the fairness factors, derived from the three-step test, has been conducted in relation to the US fair dealing exceptions above, where it was shown that the exception remains overly narrow, anti-competitive and a barrier to productivity. In these respects, the observations on the second and third factors of the three-step test above, in relation to section 19B(2), are equally applicable to a potential exception under section 12A.

Considering the dearth of local case law on exceptions to copyright,[1162] in particular on computer programs, and the lack of technical sophistication or detail in these cases, it is submitted that South Africa is not in a position to develop and sustain a fair use

---

(aa) such use serves a purpose different from that of the work affected; and

(bb) it is of a commercial nature or for non-profit research, library or educational purposes; and

(iv) the substitution effect of the act upon the potential market for the work in question.

(c) For the purposes of paragraphs (a) and (b) the source and the name of the author shall be mentioned."

[1162] Notable cases which deal with exceptions to copyright protection are *South African Broadcasting Corporation SOC Ltd v Via Vollenhoven and Appollis Independent CC and Others* 2016 4 All SA 623 and *Moneyweb (Pty) Limited v Media 24 Limited and Another* 2016 (4) SA 591 (GJ). However, neither case dealt with computer programs or provided any guidance on how fair use would be applied in the South African context. See the discussion of salient points in the *Vollenhoven* case at fn 1172 below.

exception on decompilation. There is a real risk that, if a court does grant a transformative-use exception, this will do great harm to the economy by creating uncertainty about the protection of computer programs. The potential for harm is clearly far greater than that which may result, if any, from the introduction of a statutory exception, even without an interoperability exception.

In light of the emphasis that article 6(3) of both Software Directives[1163] place on the third factor, namely, unreasonable prejudice, of the three-step test, when a decompilation exception is made, a fair use exception is likely to fail the test for, at least, the following reasons.

First, the doctrine of separation of powers[1164] oblige the court to restrain from granting wide exceptions or limitations on statutory rights unless mandated to do so by the Constitution. Considering that section 19B(2) already makes exception for decompilation, any additional exception is likely to be equally narrow. This will lead, at best, to a *de novo* motion for fair use in relation to every other reason for decompilation. In light of section 19B(1), it is also likely that the court could be persuaded to insist that the programmer first seek other, less invasive, means to achieve their goal, before approaching the court.[1165] Furthermore, strong academic

[1163] Council Directive on The Legal Protection of Computer Programs; Directive of the European Parliament and of the Council on the Legal Protection of Computer Programs.

[1164] In *National Coalition for Gay and Lesbian Equality and Others v Minister of Home Affairs and Others* [1999] ZACC 17; 2000 (2) SA 1 at [66] the court stated that the doctrine "involves restraint by the courts in not trespassing onto that part of the legislative field which has been reserved by the Constitution, and for good reason, to the legislature." In *National Treasury and Others v Opposition to Urban Tolling Alliance and Others* (CCT 38/12) [2012] ZACC 18; 2012 (6) SA 223 (CC) at [44] the court held that "courts in turn must refrain from entering the exclusive terrain of the Executive and the Legislative branches of Government unless the intrusion is mandated by the Constitution itself." More recently, the separation of powers doctrine was confirmed and applied by, among others, the court in *Lawyers for Human Rights v Minister of Home Affairs and Others* (CCT38/16) [2017] ZACC 22; 2017 (5) SA 480 (CC) at [65]-[67] regarding the remedy of reading-in. See also Currie I et al *The New Constitutional and Administrative Law* (2001) 289 for a discussion of the doctrine and the justiciability of socio-economic rights.

[1165] Section 19B9(1) creates an impression that there are other means to discover the ideas which do not involve decompilation. In US case law the court refused to impose a duty on the programmer to use opinion suggests that a limitation on intellectual property rights should also be measured for its potential to deprive the owner of his property right, before a limitation may be justified in law.[1166]

Second, the constitutional perspective on copyright as property[1167] will oblige the court to conduct a balancing exercise of fundamental rights when interpreting legislation,[1168] which is clearly eschewed toward the interests of property rights holders.[1169] Thus,

---

the least efficient means of reverse engineering. See the discussion of the *Sega II* judgment in paragraph 4 3 2 9 above. In terms of South African law, this will require, at least, that an order instructing a proprietor to permit decompilation be technically specific, because an "order must not be so general that [the] target would not know what is required of them." See Currie et al *The New Constitutional and Administrative Law* 289 fn91.

[1166] Shay R M and Van der Walt A "Constitutional Analysis of Intellectual Property" 2014 *Potchefstroom Electronic Law Journal* 17 (1) 52 at 55. Although this opinion relates primarily to review of legislative measures which may amount to an expropriation of property in terms of the dicta in *First National Bank of SA Limited t/a Wesbank v Commissioner for the South African Revenue Services and Another; First National Bank of SA Limited t/a Wesbank v Minister of Finance* (CCT19/01) [2002] ZACC 5; 2002 (4) SA 768 (*FNB*), the fact remains that it has established precedent for the argument that any limitation of property rights, even if developed through case law, must be tested in light of these factors before it may be granted. See further the methodology for considering a new fair dealing exception proposed by Shay R M 2012 *Users' Entitlements under the Fair Dealing Exceptions to Copyright* Thesis: LLM Stellenbosch University para 5.2.3 *et seq*., which relies heavily on the three-step test and judgments in *FNB* and *Laugh It Off Promotions CC v South African Breweries International (Finance) BV t/a Sabmark International and Another* 2006 (1) SA 144 (CC) (*Laugh It Off*).

[1167] *Certification of the Constitution of the Republic of South Africa, 1996* (CCT 23/96) [1996] ZACC 26; 1996 (4) SA 744 (CC) at [75] where the court refused to recognise intellectual property as a fundamental right. See Dean O H "The case for the recognition of intellectual property in the Bill of Rights" 1997 *Journal of Contemporary Roman Dutch Law* 60 105 at 107 where the author submits that the judgment in the Certification case must be read in light of the opinion of the Constitutional Assembly that "protection of IP in the Bill of Rights would be adequately addressed by a property clause entrenching rights of ownership in property." While this may be said to give some effect to South Africa's obligation in terms of article 15(3) of the International Covenant on Economic, Social and Cultural Rights (to secure the right to "benefit from the protection of the moral and material interests resulting from any scientific, literary or artistic production of which he is the author"), it does not give full effect to article 15(1) or (2): the rights of *everyone* to "take part in cultural life" and "enjoy the benefits of scientific progress and its applications".

[1168] *Makate v Vodacom (Pty) Ltd* 2016 (4) SA 121 (CC) at 87.

[1169] See the discussion in fn1164 above and accompanying main text.

decompilation will have to fit within the right to freedom of expression. In this respect, precedent[1170] will suggest that an exception to an intellectual property right should only be made where there is no clear indication of economic harm. As a result, the court is more likely to grant a transformative-use exception only if guarantees are put in place that no economic harm will result from the activity. This is confirmed in another case, which held that copyright protection is primarily a measure to safeguard economic interests.[1171] This additional burden may well render the exception inoperable as a tool for development.

Third, in order to construct a transformative-use exception, the court will have to deal with precedent that suggests copyright law is not primarily concerned with facilitating the distribution of ideas and that a "public good or welfare" reason is not sufficient to create a new exception to copyright law.[1172] Once again, the constitutional status of copyright as property will hamper the application of fair use based on public policy because, as the court held:

> "Caution should be exercised in elevating lofty pronouncements to guiding principles in ascertaining the intent and purport of the Act within our constitutional framework."[1173]

Clearly, South African courts are ill-prepared to entertain a fair use regime. As the court held, "the view that copyright aims to promote public disclosure and dissemination of works cannot be regarded as a true reflection of the purpose or intent of the Act and is not part of our copyright law."[1174] This in in stark contrast to the regime from which section 12A is copied, namely, the US, where the primacy of the public interest in copyright and, in particular, the duty to permit and promote dissemination

---

[1170] *Laugh It Off Promotions CC v South African Breweries International* at 102.

[1171] See *South African Broadcasting Corporation SOC Ltd v Via Vollenhoven and Appollis Independent CC and Others* [2016] 4 All SA 623 (*Vollenhoven*) at 29 where the court held:

> "[The] main purpose of the Act […] is to grant a qualified monopoly to the copyright owner and to reward the creator of the intellectual property."

[1172] *Vollenhoven* at 25.

[1173] At 28.

[1174] At 30.

of work, is constitutionally enshrined *and mandated*.[1175] Thus, before the application of section 12A in South African law can take place, a court will have to substantially amend the constitutional perspective on intellectual property in general *and* precedent on the nature of copyright law.

There is also a danger that the flexibilities of the three-step test may be nullified where it is codified in national law, turning the test into an additional economic impact test which each exception or limitation must overcome.[1176] The South African Copyright Act has done so, twice. The wording of the three-step test is repeated in section 13 as the basis of vesting ministerial powers to make additional exceptions in regulations, while the power to make exceptions to the right to reproductions is further limited to cases where "the cumulative effect of the reproductions does not conflict with the normal exploitation of the work to the unreasonable prejudice of the legal interest and residuary rights of the author."[1177]

Quite apart from the fact that this power has not been used since shortly after the promulgation of the principal act, the transposition of the three-step test in national law poses the very real risk that a court will not apply the flexibilities of the three-step test if it continues to view copyright as primarily intended to safeguard economic interests. There is no reason to submit that the court should ignore a criterion for making exceptions which is made expressly incumbent on a member of the executive. The opposite effect if far more likely.

[1175] See the discussion in paragraphs 4 3 2 9-10 above.

[1176] As Geiger et al *American University International Law Review* at 617 submit:

> "[While] the three-step test can be interpreted as a flexible policy instrument, *the transposition of the international three-step test into national law can fundamentally modify its operation*. Specifically, when the three-step test is implemented in national law as an *additional* control mechanism with regard to E&Ls that have already been defined narrowly, the test is no longer performing the enabling function it has at the international level." (original emphasis).

[1177] Regulations to the Copyright Act, section 2(b).

5 4 2 The impact of technological protection measures

In addition to the above obstacles, the 2017 Copyright Amendment Bill will also introduce a complex set of digital rights management (DRM) measures,[1178] divided between provisions regarding technological protection measures (TPMs)[1179] and copyright management information.[1180]

The anti-circumvention measures regarding TPMs in the Bill contain largely the same wording as that of the DMCA and the CDPA, with two important differences.

First, like the CDPA and the DMCA, section 27(5A) contains the same trafficking prohibitions regarding manufacture of, and dealing in, circumvention devices or providing information about, or assistance in, using such devices.[1181] Unlike the CDPA, but in concert with the DMCA, section 27(5A) also contains a prohibition on the act of circumvention[1182] where a TPM has been applied to a computer program.

[1178] Section 27(a) of the Bill, adding section 27(5A) to the Copyright Act and section 29 of the Bill adding sections 28O and 28P.

[1179] Section 27(5A)(a) and (b).

[1180] Section 29 of the Bill adding sections 28R and 28S to the Copyright Act.

[1181] Section 27(5A)(a)-(b). The whole of section 27(5A) reads as follows:

> "(5A) Any person who, at the time when copyright subsists in a work that is protected by a technological protection measure applied by the author or owner of the copyright -
>
> > (a) makes, imports, sells, distributes, lets for hire, offers or exposes for sale or hire or advertises for sale or hire, a technological protection measure **circumvention device** if -
> >
> > > (i) such person knows, or has reason to believe, that that device will or is likely to be used to infringe copyright in a work protected by a technological protection measure;
> > >
> > > (ii) such person provides a service to another person to enable or assist such other person to circumvent a technological protection measure; or
> > >
> > > (iii) such person knows or has reason to believe that the service contemplated in subparagraph (ii) will or is likely to be used by another person to infringe copyright in a work protected by a technological protection measure;
> >
> > (b) **publishes information** enabling or assisting any other person to circumvent a technological protection measure with the intention of inciting that other person to unlawfully circumvent a technological protection measure in the Republic, or
> >
> > (c) **circumvents such technological protection measure** when he or she is not authorized to do so,
>
> shall be guilty of an offence and shall upon conviction be liable to a fine or to imprisonment for a period not exceeding five years, or to both a fine and such imprisonment" (emphasis added).

[1182] Section 27(5A)(c).

This brings the whole of the conflict between the scope of copyright protection and the idea/expression dichotomy, as analysed above in relation to the DMCA, into South African law and is a departure from the close association between the Copyright Act and the CDPA.

Second, unlike both the US and UK positions, all the anti-circumvention measures in the Bill are statutory *criminal* offences, not instances of copyright infringement. This exacerbates, exponentially, the already unfair and unjustifiable limitation which TPM measures place on the utility of a decompilation exception *per se* and, in particular, the lawful act of decompilation for a reason other than interoperability. The Bill does include a safeguard against the basic impact of the anti-circumvention measures, similar to the DMCA and the CDPA, by providing that:

> "For the purposes of this Act and of section 86 of the Electronic Communications and Transactions Act, 2002 (Act No. 25 of 2002), **nothing in this Act shall prevent** any person from using a technological protection measure circumvention device to perform any of the following:
>
> > (a) **An act permitted in terms of any exception** provided for in this Act; or
> >
> > (b) the sale, offer to sell, procurement for use, design, adaptation for use, distribution or possession of any device or data, including a computer program or a component, which is designed primarily to overcome security measures for the protection of data in order to enable the performance of any act permitted in terms of paragraph (a)."[1183]

However, the copy-and-paste drafting of this provision has the same effect of re-enforcing the interoperability limitation as it does under US and UK law and it applies only to fair dealing, not to fair use.[1184] Despite the widely-proclaimed pro-developmental stance of the Bill, no attempt has been made to temper the manifestly unfair nature of the TPM provisions or give effect to the flexibilities offered by the three-step test to create wider exceptions for developmental goals.

[1183] Section 28P (emphasis added).

[1184] Section 29P(a) refers only to exceptions "provided for in this Act", not to exceptions developed in terms of, or in relation to, the Act, such as fair use.

On the contrary, the curiously draconian *criminalisation* of circumvention appears to be primarily intended to serve, and secure, the beneficial interests of existing, and predominantly foreign, software owners.[1185] This is made clear by the Bill in relation to the minister's power to declare, by regulation, permitted acts of circumvention,[1186] with exactly the same wording which vest this power in the Librarian of Congress in terms of the DMCA.[1187] When doing so the minister must, inter alia, consider "the effect of the circumvention of technological protection measures **on the market for or value of works** protected by copyright."[1188] Considering that the minister has not made any new regulations since 1978, there is no reason to assume that South Africa will be able to maintain a system of exceptions and limitations resembling that of the US.

A further indication of this conclusion is the fact that, despite extensive commentary and strong academic opinion to the contrary,[1189] the Bill sustains the continued application of the additional criminal offences in the Electronic Communications and Transactions Act[1190] regarding circumvention of measures that protect data.[1191]

In addition, the indications in SA case law, that a balancing exercise will favour a protectionist approach to exceptions in copyright law,[1192] is a further source of concern

---

1185 As Van der Merwe et al *Information and Communications Technology Law* 337 point out: "the balance has been tilted in favour of authors in an attempt to help them cope with some of the threats arising from new technologies."

1186 Section 39(cH).

1187 Section 1201(a)(1)(C).

1188 Section 39(cH)(iv).

1189 Jooste C and Karjiker S "Intellectual Property Law in the Digital Environment (EIP Law)" 443-4; See also Dean O H, et al. "Commentary on the Copyright Amendment Bill 2015" http://blogs.sun.ac.za/iplaw/files/2015/08/CIP-Formal-Comments-Copyright-Amendment-Bill-2015-Online1.pdf) at 59-61 (discussing the previous draft of the Bill); Jooste C and Karjiker S "Commentary on the Copyright Amendment Bill 2017" http://blogs.sun.ac.za/iplaw/files/2017/06/CIP-Comments-Copyright-Amendment-Bill-2017.pdf) at 7, 19, 38-42 (regarding the first edition of the 2017 draft).

1190 Electronic Communications and Transactions Act 25 of 2002. See Van der Merwe et al *Information and Communications Technology Law* 79-82 for a discussion of these provisions.

1191 Particularly section 86(4):

> "A person who utilises any device or computer program mentioned in subsection (3) in order to unlawfully overcome security measures designed to protect such data or access thereto, is guilty of an offence."

1192 See the footnotes on the *Vollenhoven* case above.

regarding the application of section 28P. The second factor of the three-step test already contains a strong disposition toward the current *and future* economic rights of the author. Considering that the majority of decompilation cases dealt with making potentially functionally-similar or complementary works, which may have an impact on future revenue, and the extended reading of a right in copyright afforded by section 28P, there is a real likelihood that the anti-circumvention measures will make it impossible to access code for anything other than interoperability, defined as narrowly as possible.[1193] This conclusion is reinforced by the Bill, which permits incidental or transient copying or adaptation that is essential to carry out a technical process (i.e. decompilation) only if it has "no independent, *economic* significance,"[1194] limits personal use (the basis of a fair use decompilation exception) to non-commercial purposes[1195] and imports the "substitution effect [on] the potential market"[1196] as criteria for fair use.

The pro-developmental spirit of the Bill and South Africa's policy objectives stand in stark contrast to every proposed provision to update copyright law in relation to computer programs since 2015. Whether considered individually or collectively, these provisions are not fair and do nothing other than enshrine the unfortunate consequences of a literary-analogy and the inappropriate advantages this affords copyright owners.

Considering that the confines of the literary-analogy have been shown as the root cause of all of the unfortunate limitations on software development discussed in this work, it is lamentable that the legislature has failed to take advantage of the flexibility offered by *sui generis* classification in SA law, or, at least, the freedom offered by the three-step test, as a way to set legal development on a path that is better aligned with

---

[1193] As Geiger et al *American University International Law Review* at 595 points out:

> "Bearing in mind the possibilities for right holders to control the uses of their works through technical measures, in the long run, this [future economic interests] could even significantly restrict E&Ls in the digital environment."

[1194] Section 12C (emphasis added).

[1195] Section 12B(i).

[1196] Section 12A(iv).

national policy, local market conditions and the true technical nature of computer programs.

The “enabling” role of the three-step test, which allows “social, cultural, and economic interests […] to be balanced against the rationales of copyright protection”[1197] has successfully been used by several courts in, inter alia, Germany, Switzerland, Colombia and Spain,[1198] in addition to those in the US discussed above, to grant, or extend, wider exceptions in the public interest. It is lamentable that the South African position has not only failed to do the same but also made it very difficult, if at all possible, for the courts to do so.

Instead, the future of decompilation and copyright law in SA seems to suggest that attempts to resolve the idea/expression dichotomy will continue to favour the copyright owner, at least insofar as non-literal elements and underlying ideas are concerned.

5 4 3 The impact of contractual terms

Unlike the CDPA, which safeguards the decompilation exception against being excluded in a EULA, section 19B of the Bill does not contain a comparable provision to render such a term void.

Instead, the Bill proposes a general voidability clause:

> “To the extent that a term of a contract purports to prevent or restrict the doing of **any act** which by virtue of this Act would **not infringe copyright** or which purport to renounce a right or protection afforded by this Act, such term shall be **unenforceable**.”[1199]

While the first part of this provision may be read to allow circumvention for the purpose of decompilation in terms of the statutory exception, this conclusion is not certain. First, this section refers to acts which do “not infringe copyright” – considering that the act

---

[1197] Geiger et al *American University International Law Review* 618.

[1198] See Geiger et al *American University International Law Review* 618-622 for a discussion of these cases, all of which related to works or rights not related to computer programs. For this reason, the cases are not analysed in this work.

[1199] Section 38B(1) (emphasis added).

of circumvention is a criminal infringement of copyright, the real possibility exists that the voidability clause will not apply to acts of circumvention, leaving the proprietor free to use the TPM measures as an alternative basis for copyright infringement similar, but worse,[1200] to that of the US.

Second, unlike the CDPA, the voidability clause renders certain contractual terms unenforceable, not void. That means it reserves the option to challenge, and uphold, the legality of a contractual term which prohibits decompilation under certain circumstances. Furthermore, it safeguards the application of a decompilation prohibition in the EULAs of all US software made available in South Africa, until it is successfully challenged in a local court. In the South African legal context, this is unlikely to occur.[1201]

Even if the significant burdens associated with access to justice are overcome by a local developer seeking to circumvent a TPM just to legally decompile the work, a court's finding on this point will be fact-dependant, because the existence of such a term remains only voidable, not void. This means each developer must challenge the terms of the licence individually for each program.

Practically, there is no way to view this other than as an absolute prohibition on the decompilation of all software, with a significant advantage to the largest software exporting nation in the world.

[1200] It will be worse because the measure is applied as a criminal sanction in South Africa.

[1201] Considering that, in order to succeed with an application that a court should not enforce a contractual provision in the public interest, it must be shown that enforcement would be unreasonable *inter partes*, this will pitch the established view of the courts in favour of protecting the economic interests in copyright against the limited public interest argument of the applicant. It is submitted that in the majority of cases the motion will be dismissed and the contractual term upheld. For a discussion of the test applied by the court, see Kerr A *The Principles of the Law of Contract* 6ed (2002) 663. See also at 665 where the author submits that "sphere of operation of the contract" is the guiding principle in enforcement disputes. In the case of software licenses, it may well be argued that a restriction falls within the economic purpose of the contract.

This raises alternative arguments regarding the enforceability of the EULA in light of consumer interests[1202] and competition law[1203] - a full analysis of which falls outside the scope of this work.[1204]

[1202] In particular the right to fair, just and reasonable terms and conditions in section 48 of the Consumer Protection Act 68 of 2008 (CPA):

> "(1) A supplier must not—
>
> (a) offer to supply, supply, or enter into an agreement to supply, any goods or services
>
> (i) at a price that is unfair, unreasonable or unjust; or
>
> (ii) **on terms that are unfair, unreasonable or unjust**;
>
> (b) market any goods or services, or negotiate, **enter into** or administer a transaction or an **agreement** for the supply of any goods or services, in a manner that is unfair, unreasonable or unjust; or
>
> (c) **require a consumer**, or other person to whom any goods or services are supplied at the direction of the consumer
>
> (i) **to waive any rights**;
>
> (ii) assume any obligation; or
>
> (iii) waive any liability of the supplier,
>
> **on terms that are unfair, unreasonable or unjust**, or impose any such terms as a condition of entering into a transaction."

[1203] The competition issues associated with restrictive software licensing is fully canvassed in chapter 5 of Stokes *Digital Copyright: Law and Practice*.

[1204] See Shemtov *Beyond the Code* 19-23 for a comprehensive analysis of the imbalance between the public and private interests created and maintained by software licenses. See also the comprehensive analysis of measures which may be used to counteract this effect in chapter 2 of the same work. In the South African context, see Mupangavanhu Y "Fairness a slippery concept: The common law of contract and the Consumer Protection Act 68 of 2008" 2015 *De Jure* 48 (1) 116 at 121, highlighting the fact that contracts of adhesion, such as a EULA, potentially undermine freedom of contract, and 129 et seq. discussing the CPA as a means to promote fairness in contracts. See also Barnard J and Mišćenić E "The role of the courts in the application of consumer protection law: A comparative perspective" 2019 *Journal for Juridical Science* 44 (1) 111 116-7 discussing the South African position regarding consumer protection and the limited effect it may have where the agreement also contains an arbitration clause – something which is common to software license agreements. Regarding the alternative consumer protection measures in the Electronic Communications and Transactions Act, see generally Jacobs W "The Electronic Communications and Transactions Act: Consumer Protection and Internet Contracts" 2004 *SA Merc LJ* 16; Snail S L "South African e-consumer law in the Context of the ECT Act (part 1)" 2007 *The Quarterly Law Review for People in Business* 15 (1). None of the consumer protection measures of the ECT Act are applicable to contractual prohibitions on decompilation. These measures deal primarily with the duty on a web-trader to provide information, and limited rights of return of the consumer.

5 5 A fair decompilation approach

It has been made clear from the outset that it is the purpose of this work to advocate for an alternative approach to the protection of computer programs that leverage its status as a *sui generis* type of work. In this context, decompilation is the area of primary concern for a *normative* analysis of the idea/expression dichotomy and the extent to which the prohibition on decompilation facilitates the protection of non-literal elements as a result of the restriction on access to code.

Therefore, it is inappropriate, in the context of a normative analysis, to suggest specific legislative amendments. It has also been shown that reliance on foreign law is clearly inappropriate in this context. Furthermore, in light of the unenacted and widely-criticised status of the South African provisions on this point, it would be academically unsound to attempt to make suggestions for statutory reform in any detail. Any proposal would be, at best, speculative and may yet be rendered moot by future changes to the bill.

However, as a means to draw together, and illustrate the potential of, the research results of this work, one simple suggestion may be made about an approach to legislative regulation of decompilation which is both academically and technologically sound.

It is suggested that a decompilation exception should be developed based on the principle of access to the limited teaching of the code. This has been alluded to above as a *reading exception* for computer programs. In this context, the word ‘read’ is used to refer to the usual act of textual interpretation carried out by humans, not to the act of ‘reading’ a program where it describes the action of executing a program on a computer.

In the simplest terms, this means that a statutory decompilation exception, like that in section 19B, could become a reading exception by minor alteration. For the sake of international harmonisation, it might even retain the technically inaccurate and unnecessary references to translation and adaptation, provided that it is altered to replace the interoperability limitation. It may also retain the indispensable standard. Neither criteria would limit the utility of the exception.

Thus, for example:

(1) The authorization of the copyright owner shall not be required where reproduction of the code and translation of its form are indispensable in order to obtain the information necessary to ~~achieve the interoperability of an independently created computer program with other programs~~ **understand the operation of that program** if the following conditions are met:

> (a) The acts are performed by the licensee or another person having a right to use a copy of the program, or on their behalf by a person authorized to do so;
>
> (b) the information necessary to ~~achieve interoperability~~ **understand the operation of the program** has not previously been readily available to the persons referred to in paragraph (a); and
>
> ~~(c) those acts are confined to the parts of the original program which are necessary in order to achieve interoperability.~~

Formulated in this way, the exception places no greater limitation on the rights of the copyright owner than it does currently. The possibility to pursue an infringement action based on literal reproduction remains unaffected. So too does the limited options of successfully claiming infringement in SSO or functionality. Nothing in this wording could be misread to provide an avenue for piracy, because the law regarding textual reproduction remains the same.

It is not the extent or the scope of permitted reproduction that is changed by this alteration, only the purpose for which it is warranted. The only change is that the person carrying out the decompilation process may do so for any lawful reason, not just the limited interoperability purpose currently permitted. This has the wide-reaching effect of removing any constitutional challenge to the protection of ideas, or arguments about the inappropriate scope of protection in computer programs.[1205]

[1205] Examples of such arguments are that the constitutional property rights of the copyright owner have been eroded by the proposed exception which narrows the scope of protection or by its effect that would result in the creation of pirated works. See further the discussion at fn 1144-9 above. Other examples of a constitutional challenge in this context include the right to access to information in terms of the right to freedom of expression and the overbroad application of restrictions on access to works discussed in para 4 2 4 1 1 above.

It may be possible to use alternative wording to replace the “understand” criteria with an “access” rationale, which would be more akin to the original decompilation exception developed in US case law. However, this would require extensive addition to the conditions in the above provision to define the *purpose* for which access is necessary, creating a circular effect leading back to a limited exception which would still fail to address the nature of the work.

It may also be possible to use an “intelligible” or “humanly legible” standard instead. However, this would require that the exception permit the manipulation of the code into a “version” that is intelligible. As shown above, decompilation does not create versions or derivations of the code. Thus, this formulation would be technically incorrect and misleading.

It is possible to argue that the “understand” criteria is open to abuse by, for example, permitting repeated actions of decompilation under a variety of technical circumstances, such as, using several different decompilers. However, this is not an argument unique to the reformulated exception. Nothing in the interoperability-version restricted the number of times the action may be carried out or the technological means which may be used. The term “understand” introduces nothing new, except a concept that is not familiar to copyright legislation. However, it is not strange for copyright law to define a wide variety of uses and, if necessary, the term “understand” may be defined by the Act or the section as: “the ability to comprehend or interpret the meaning conveyed by the expression.” If the programmer fails to understand the code, the exception offers no further recourse and thus places no greater or additional limitation on the owner’s rights.

Lastly, in order to make this exception operable, it will be necessary to amend the way in which anti-circumvention provisions are made applicable to computer programs. At the very least, it is proposed that the prohibition on the *act* of circumvention should not be applicable to computer programs protected by TMPs at all, following the guidance of the CDPA in this respect. Other matters, such as the contractual exclusion of this exception in a EULA and the status of transient copies under the Bill remain unaffected, and are matters for separate consideration outside the scope of this work.

Needless to say, the analysis of the three-step test above makes it clear that the reformulated exception will not only pass all three stages of the test but will also be manifestly fair when measured against the normative standards inherent in each step. In addition, the minor alteration has a significant effect on leveraging the flexibilities of the test and gives direct, clear, legally enforceable and unobjectionable effect to South Africa's national development goals and policy position on socio-economic reform to copyright law, without any risk of violating international law or endangering the protection of local work abroad. Finally, it is submitted that this approach to decompilation is in full accord with the spirit and objectives[1206] of the WCT[1207] and may, therefore, also be considered a means to ensure alignment of national law with the WCT.[1208]

[1206] In particular the following, stated in the preamble to the WCT (emphasis added):

> "Recognizing the **profound impact of** the development and convergence of information and communication technologies **on the creation and use** of literary and artistic works.
>
> […]
>
> Recognizing the need to **maintain a balance** between the rights of authors and the larger public interest, particularly education, research and **access to information**, as reflected in the Berne Convention"

And the agreed statement concerning article 10:

> "[…] Similarly, these provisions should be understood to permit Contracting Parties to devise **new exceptions and limitations that are appropriate in the digital** network environment."

[1207] In the same way, and to the same extent, that it is permissible to make exceptions that are unique to the policy demands of a nation and peculiar to the uses of work in the digital environment in terms of TRIPS, as discussed in detail in paragraph 5 2 2 above.

[1208] In particular, the proposed exception relies on the fact that the WCT, intentionally, did not prohibit the temporary reproduction of work where it is necessary to perceive the work or reproduced during lawful use of the work. The wording of the WCT omitted reference to incidental reproductions based on a proposal by the South African delegation to the diplomatic conference and the negotiations on the text. See WIPO 1999 *Records of the Diplomatic Conference on Certain Copyright and Neighboring Rights Questions: Summary Minutes of Main Committee I* Geneva 670-1. See also Wang J *Conceptualizing Copyright Exceptions in China and South Africa* (2018) 108-9.

Chapter 6

Conclusions

6 1 Introduction

This work has made some effort to disentangle computer programs from literary works when the application of copyright law to the act of decompilation is considered. This has aided in clarifying how, and why, the research findings, summarised below, are justified, based on both a legal and technical analysis of the type of work.

In this way, although it has been critical of a literary-analogy in relation to computer programs, it has not dismissed the utility of an analogy as a method to illustrate the universal applicability of fundamental copyright principles. Thus, for the first and last time in this work, and for the sake of argument, the literary-analogy is used to illustrate some of the findings of this work. This is done under caution: it does not endorse the literary-analogy in relation to decompilation and should not be used to perpetuate, or give rise to, further arguments in favour of prohibiting decompilation – it is just a final attempt to convince those who would, despite the findings of this work, still insists on interpreting copyright law in relation to computer programs by analogy to literary work.

6 1 1 A literary-analogy of the conclusions

The act of compiling source code into object code is not, as was shown above, analogous to translating a book from one language into another. It is more akin to the transformation that occurs when one person turns the novel of another into a screenplay.

Based on this analogy, it should be clear that decompilation is similar to an attempt to return the screenplay to its original state in the form of a novel, based only on the wording of the screenplay. This is clearly impossible because there are certain elements of the original novel that are not present, or alluded to, in the screenplay. Thus, no matter how sophisticated the reverse-engineering process is, it will never ‘reveal’ the novel. At best, it can be said that a new novel has been written based on the screenplay. This is the extent to which decompilation is *potentially* an adaptation of the object code.

At this point, the influence of the literary-analogy must cease. It may not be allowed to suggest that, because decompilation is *similar* to adapting a screenplay into a novel, it is the *same* act when it comes to a finding on whether the work has indeed been adapted. To make such a finding would be to ignore the unique nature of the type of work. Therefore, it has been shown that the technical reality of decompilation is distinctly different from the act of translation and that a narrower view of copying, for the purpose of adaptation, is appropriate if copyright law is to avoid unduly restricting the use of ideas in computer programs.

But, the literary-analogy above has another useful purpose. It illustrates the basic difference between source and object code and the distance that it creates between the processes of compilation and decompilation. On this basis, it makes it clear why decompilation should, *per se*, not be prohibited or, where it is, permitted as widely as possible. In other words, in the same way that the person in lawful possession of a screenplay is allowed to take any steps to understand the words in that play, that person should be free to decompile object code in order to read it. For example, if the screenplay is only available in German, there is nothing in copyright law that prevents that person from using a computer program to create an English translation thereof so that it can be read. But the same cannot be said for decompilation. As a general rule, this action is considered a prohibited adaptation even though the work exists only in a ‘language’ that nobody can read. This is, to some degree, analogous to the situation where the screenplay is only made available in braille and the law prohibits the mechanical transformation thereof into alphanumeric characters.

Furthermore, where a decompilation exception is granted to allow ‘translation’ for interoperability only, it is tantamount to stating that the screenplay may only be translated if, and to the extent that, it is done in order to write a poem. The interoperability limitation is just as inappropriate to computer programs as it would be to prescribe the type of work that may be created after a screenplay, book, article or other literary work has been read.

And, in the case of computer programs, the law goes one step further to make it unlawful to attempt to see the work. Where the act of circumventing a technological protection measure, applied to a computer program, is prohibited, it is tantamount to placing all literary works in the world in a single library and allowing only paying members to go inside, or those who have the permission of the copyright owners. In this situation, the entire work is inaccessible to everyone and impossible to read, *in terms of copyright law*, despite the fact that all of the books will contain ideas that are not subject to copyright protection. The rationale for anti-circumvention measures applied to computer programs is analogous to the argument that this global library must control access to the books because, otherwise, someone might copy one of the books. While this may be true and justified in terms of copyright law as a general proposition, it is certainly overbroad, at least in the case of computer programs, where the right to read the work is already restricted by decompilation provisions.

Anti-circumvention provisions allow the copyright owner to extend his rights beyond the scope of copyright law by simply imposing an additional entry criterion in the license. In addition, if that entry criterion is breached, copyright law supports the economic interest of the library and the copyright owner only, by imposing a penalty or other sanction on anyone who tried to access the library without permission.

In this context, the literary-analogy explains why this work has sought to create room for national public policy and developmental objectives. It recognises that, in the same way that there is an expectation of access to the contents of books in a publicly-funded library, in order to read it, there are justifiable public interests that underpin the argument for removing the copyright-based restrictions on decompilation.

In the same way that putting a book in a public library does not create the risk that the book will be copied, the ability to freely decompile a program does not have an impact on piracy. The risk of the book being copied is due to the fact that the book exists. The same is true for computer programs. As long as the work is publicly available, the risk of piracy exists. Allowing the book to be read is no more responsible for piracy than permitting decompilation of a computer program would be. In addition, unlike books, there are far easier and less expensive ways of creating pirated computer programs than by means of decompilation, amendment and then re-compilation.

And, even if it could be said that decompilation may cause a rise in piracy, the principles of copyright law applicable to unauthorised reproduction remain just as effective with or without a decompilation exception – neither the scope of protection nor the test for infringement is affected by a general decompilation exception. The only thing that has changed is the extent to which the copyright owner in a computer program is entitled to argue that his work has been infringed when *a lawful copy* thereof is accessed and read.

It has also been shown that the rationale for restricting permissible reproduction, in the case of decompilation, is based solely on the commercial, competitive interests of the copyright owner in the same way that the hypothetical library exists only to generate an income for the library and the copyright owners. While both examples of economic interests are justified in terms of copyright law, it may not be the *sole* justification. This work has, hopefully, shown that public interest remains an important, if not the most important, consideration when the rights to control access to, and reproduction of, a computer program are determined.

Thus, to summarise the literary-analogy used in this chapter, it may be concluded that decompilation must be viewed, in terms of copyright law, as a *sui generis* activity akin to reading a book. Any provision that prevents or restricts the act of reading is inconsistent with the public/private interest balance and violates the idea/expression dichotomy. Insofar as decompilation is the same as trying to understand a book, it may not be regulated by copyright law. Insofar as decompilation is dissimilar to reading but akin to adaptation, copyright law should exempt this act *per se*.

Lastly, in case it has not been clear thus far, the approach of copyright law to decompilation should never be influenced by considerations about what the person could do with the results of decompilation. In other words, when deciding on the regulation of decompilation, it is irrelevant whether or not the decompiled code might be used to create derivative, competing or pirated works. The legality of the product that is created with the help of decompiled code, is the domain of copyright infringement in general, not decompilation. In this sense, the public interest in reading the code far outweighs the private interest in restricting access to an intelligible approximation of the expression.

6 1 2 The research findings

At this point, one may return to each of the research questions, stipulated in chapter one, to summarise the findings made in this thesis.

6 1 2 1 What is decompilation?

This question required both a legal and technical analysis. From a legal perspective, in terms of copyright law, decompilation is the unauthorised act of turning a copy of a protected computer program, in object code form, into a set of instructions expressed in source code form. This process requires the reproduction of the entire program, by a computer, and the automated interpretation of the work. The end result is a work that is legible and expressed in a programming language. The initial loading of the object code is a reproduction of the code. The result of the decompilation process is also considered an adaptation of the object code, either as a translation of the code or an expression of the code in a different notation, depending on the construction of copyright law in the particular jurisdiction. In the US and UK, the act of decompilation does not amount to the making of an unlawful reproduction or adaptation if the act is carried out for the purpose of interoperability. In South Africa, where no exemption is currently granted, decompilation is prohibited by operation of copyright law.

From a technical perspective, decompilation is an attempt to discover the underlying logic of, and ideas that are expressed in, the program by manipulating the mathematical calculations performed by the object code in order to estimate the source code instructions that were originally written and might have resulted in the object code. For this reason, decompilation is an automated process of interpreting the object code, according to known programming techniques and estimations about the most likely source code instructions which gave rise to it, in order to deliver a report, in legible programming language terms, about how the program could have been written prior to compilation. Decompilation creates a new set of source code instructions, based on an interpretation of the purpose and operation of the object code.

6 1 2 2 How does South African copyright law address decompilation?

The position in SA relies entirely on statutory interpretation in a broad sense, because there is no local case law on decompilation. Currently, the act of decompilation is not a form of fair dealing, or otherwise permitted by the Act. Therefore, the restricted acts of reproduction and adaptation are potentially infringed when a program is decompiled. However, there is no authority to substantiate the argument that decompilation constitutes the making of a translation. Nor is it correct to interpret SA law to the extent that decompilation is a reproduction of the original source code or an adaptation of the object code by way of a change in programming language or code.

The only basis for prohibiting decompilation is adaptation by way of a change in notation. This argument requires that the essential requirement of copying, inherent in a case of adaptation, is interpreted in the wide sense to mean use of the work to create a derivative work. It was found that local case law on the wide meaning of copying, in cases of adaptation, is only theoretically capable of being applied to decompilation, despite the precedential shortcomings of these cases in relation to computer programs, and is not a sufficient basis for a prohibition on decompilation. Consequently, it is found that, although decompilation is prohibited by copyright law in SA, the legal basis for this view is underdeveloped and insubstantial.

It is submitted that the precarious basis of the decompilation prohibition in SA copyright law, coupled with the technical nature of decompilation, justify an interpretation of copyright law which permits decompilation – to the extent that it falls outside the scope of any of the prohibited acts when a narrow view of copying is adopted.

6 1 2 3 Is this position fair and appropriate in light of foreign precedent?

The position in SA is only fair if the decompilation justification, outlined above, is adopted to permit decompilation. However, there is no comparable position in foreign law which permits decompilation in the absence of a decompilation exception. Consequently, if the wide meaning of copying is applied to computer programs in the same way it applies to other types of work, the SA position is that decompilation remains a form of infringing activity because there is currently no exemption applicable to this process. This, it is found, is manifestly unfair and amounts to the protection of ideas in relation to computer programs. The fairness finding, in this respect, is based on the analysis of statutory development and case law in the US and UK where, in both cases, it was considered essential that decompilation should, to some extent, be exempted in order to prevent the application of copyright to ideas and maintain a balance of interests.

The analysis of foreign law also identified a number of factors that influenced the decision to limit the purpose of a decompilation exception to interoperability only. These are, in order of increasing importance: the fear that decompilation may facilitate piracy of works; the need to align copyright in computer programs with the principles applicable to literary works by providing for a limitation on the right to make translations of the work; the expectation that copyright law will continue to facilitate the licensing of computer programs under circumstances dictated by the rights holder; the role of copyright in safeguarding the economic interests of the copyright owner; the ability to leverage copyright protection as a means to obtain and maintain a competitive advantage in the market and the purpose of copyright to regulate the making of derivative works without authorisation.

Conversely, the study of foreign case law identified the range of fairness factors that supported the need for a decompilation exception and ascertained the manner in which these factors were applied, along with the above criteria, to arrive at the approach to decompilation in both the US and UK. These factors are, in order of increasing importance: the fact that decompilation is a uniquely technical process that does not fit comfortably within the existing framework for exploitation of literary works; the public interest that copyright law should not be extended to ideas; the separation between copyright and patent law regarding the protection of the function of a work; the fact that object code is not legible; the need to have access to the code in order to interpret it and discover the non-literal elements and the need to create new programs that will be interoperable with existing computer programs.

In light of these findings, which illustrate that the existence of a decompilation exception is the minimum standard expected of copyright law to maintain a degree of fairness regarding the exploitation of computer programs, it is clear that the position in SA is not fair because it does not provide for decompilation by way of an exception. In addition to the above, an analysis of the anti-circumvention measures in foreign law, and associated case law, was carried out. It was found that these provisions have an inherent capacity to negatively impact the fairness balance insofar as decompilation and the protection of ideas are concerned. In this respect, it was found that the approach to decompilation advocated in this work must also take cognisance of the need for an alignment between a decompilation exception and the prohibition on circumvention of technological protection measures.

6 1 2 4 How should decompilation in South Africa be developed?

This question was addressed in two parts. First, the three-step test was identified as the only secure standard for testing the legitimacy of a current or proposed exception or limitation on copyright protection and the means by which the idea/expression dichotomy should be resolved in a particular case. This required a close analysis of the status of the three-step test in international law, particularly in terms of international instruments developed after the introduction of the three-step test in the Berne Convention. Each step of the test was analysed, according to the most authoritative sources on the substantive interpretation of each step of the test, and it was found that the test itself, as well as each of the individual steps, are to be understood as inherently flexible standards that are capable of accommodating national policy and local market considerations which, in fact, encourage the customisation of national copyright law in order to meet technological challenges and new forms of exploitation.

Second, based on all of the above findings, an approach to decompilation was formulated. The essence of this approach, which is based on the public need to be able to take any steps necessary to *understand* computer code, is expressed by way of a proposed decompilation exception for SA copyright law. The proposal relies on the fairness factors identified in foreign law, particularly the need to have access to the ideas, and the fairness considerations embodied in the three-step test which provide flexibility for a new exception to be made that could address the progressive national policy objectives identified in chapter 2.

On this basis, it is found that it would be both fair and justifiable to implement a decompilation exception that is not limited to the purpose of achieving interoperability. As a consequence, this work proposes that the most effective and appropriate way to address decompilation in SA and, at the same time, give effect to the national policy objectives of stimulating a local economy and providing greater access to information, would be to permit decompilation where it is carried out in order to *understand* the work.

6 1 2 5 The core research question

The core research question was expressed in chapter one as follows:

Does copyright law protect ideas because of the way in which the decompilation of computer programs is regulated, and is this position fair?

In short, this work has shown that the answer to the first part of this question is **yes**, and the answer to the second part is **no** – where decompilation is concerned, copyright law does protect ideas, and it is not fair to do so. This is shown to be true where copyright law does not permit decompilation at all, like SA, and also true where a decompilation exception does exists, under either a fair use of fair dealing regime.

The reasons advanced for this finding is twofold. First, a direct and unamended application of the copyright law principles regarding adaptation and reproduction does not acknowledge the peculiar nature of the work or the *sui generis* form of use that decompilation represents. Consequently, computer programs are misunderstood, based mainly on a literary-analogy approach to computer programs, which led to the position that decompilation is an activity which must be accommodated within the scope of one or more of the restricted acts. Second, this is unfair because it prevents a user from accessing the underlying ideas, the non-literal elements, of the work. This amounts to a *de facto* extension of copyright to ideas, which is inherently unfair.

Furthermore, where an exception is made to permit decompilation, but that exception is limited by an interoperability condition, the legal position remains unfair. This is because the exclusion of ideas from the ambit of copyright must be universally unfettered. There is no basis to suggest that ideas, expressed in copyrighted work, may only be used for a specified purpose. Consequently, a limited exception based on the purpose of the use does not correct the fairness imbalance that is created by the general prohibition on decompilation.

6 2 Concluding note

In light of the above, this conclusion chapter must return to the title of this work, namely, 'decompilation and copyright in ideas - the protection of non-literal elements of computer software and the idea/expression dichotomy.'

Collectively, the findings made in relation to the core research question and all of the sub-questions, leads to the conclusion that copyright does exist in ideas, or the non-literal elements of computer programs, because of the manner in which decompilation is viewed and the manner in which it is regulated, and that this is a violation of the idea/expression dichotomy and its most fundamental ideals.

This work has sought to point out why this is so from both a legal and technical perspective on decompilation and it is hoped that this work may contribute to a better, more nuanced understanding of copyright in computer programs. In addition, this work formulated an alternative legal approach to the regulation of decompilation which, it is envisioned, may be useful to guide legal development. Throughout, the opportunity to conduct research on this matter has been used to advance a legally sound methodology for accommodating national developmental policy and public-interest consideration within the local and international copyright regime.

Acronyms

API	-	Application Programming Interface
CDPA	-	Copyright, Designs and Patents Act 1988
CJEU	-	Court of Justice of the European Union
CONTU	-	Commission on New Technological Uses of Copyrighted Works
DMCA	-	Digital Millennium Copyright Act 1998
DNPIP	-	Draft National Policy on Intellectual Property 2013
ESC	-	Economic, Social and Cultural
ESCIA	-	The WIPO Economic, Social and Cultural Impact Assessment
EULA	-	End-user License Agreement
GDP	-	Gross Domestic Product
ICESCR	-	The International Covenant on Economic, Social and Cultural Rights
IP	-	Intellectual Property
NDP	-	National Development Plan 2030
ROM	-	Read-Only Memory
SA	-	Republic of South Africa
SME	-	Small and Medium-sized Enterprises
SSO	-	Style, Structure and Organisation
TPM	-	Technological Protection Measure
TRIPS	-	Agreement on Trade Related Aspects of Intellectual Property Rights
UK	-	United Kingdom of Great Britain and Northern Ireland
UNESCO	-	United Nations Educational, Scientific and Cultural Organisation
US	-	United States of America
USC	-	United States Code
WCT	-	WIPO Copyright Treaty
WIPO	-	World Intellectual Property Organization
WPPT	-	WIPO Performances and Phonograms Treaty
WTO	-	World Trade Organization

Bibliography

Books

Armstrong C, De Beer J, Kawooya D, Prabhala A & T Schonwetter (eds.) *Access to knowledge in Africa: The role of copyright* (2010) UCT Press.

Bainbridge D I *Intellectual Property* 8ed (2010) Pearson

Bainbridge D I *Legal Protection of Computer Software* 5ed (2008) Tottel

Barfield W & Blitz M J (eds.) *Research Handbook on the Law of Virtual and Augmented Reality* (2018) Edward Elgar

Bently L and Sherman B *Intellectual Property Law* 4ed (2009) Oxford University Press

Bowrey K, Handler M & Nicol D (eds.) *Emerging Challenges in Intellectual Property* (2011) Oxford University Press

Caddick N, Davies G & Harbottle G *Copinger & Scone James on Copyright* 17ed (2019) Sweet & Maxwell

Currie I, De Waal J, De Vos P, Govender K & Klug H *The New Constitutional and Administrative Law* (2001) Juta

Dean O H and Karjiker S *Handbook of South African Copyright Law* 14ed (2012) Juta

Dean O H and Dyer A (eds.) *Introduction to Intellectual Property Law* (2014) Oxford University Press

Dworking G and Taylor R D *Blackstone's Guide to the Copyright, Designs and Patents Act 1988: The Law of Copyright and Related Rights* (1989) Oxford University Press

Edgar S L *Morality and Machines: Perspectives on Computer Ethics* (2002) Jones & Bartlett

Eilam E *Reversing Secrets of Reverse Engineering* (2005) Wiley Publishing

Galler B A *Software and Intellectual Property Protection: Copyright and Patent Issues for Computer and Legal Professionals* (1995) Greenwood Publishing Group

Kerr A *The Principles of the Law of Contract* 6ed (2002) LexisNexis

Klopper H, Pistorius T, Rutherford B, Tong L, Van der Spuy P & Van der Merwe A *Law of Intellectual Property in South Africa* (2011)

Lai S *Copyright Protection of Computer Software in the United Kingdom* (2000) Hart

Litman J D *Digital Copyright* (2006) Prometheus

MacQueen H, Waelde C, Laurie G & Brown A *Contemporary Intellectual Property Law and Policy* 2ed (2008) Oxford University Press

Metheny M *Federal Cloud Computing - The Definitive Guide for Cloud Service Providers* 2ed (2017) Elsevier

Nimmer D and Nimmer M B *Nimmer on Copyright Publication 465 Release 108 August 2019* (1978) Matthew Bender

Patry W F *Copyright Law and Practice* 2ed (1994) BNA Books

PWESCR (Programme on Women's Economic, Social and Cultural Rights) *International Covenant on Economic, Social and Cultural Rights: A Handbook* (2015) Systems Vision

Reed C & Angel J *Computer Law: The Law and Regulation of Information Technology* 6ed (2007) Oxford University Press

Ricupero R & Ortiz R M (eds.) *Resource Book on TRIPS and Development UNCTAD-ICTSD* (2005) Cambridge University Press

Scott M D *Scott on Information Technology Law* 3ed (2019) Wolters Kluwer

Senftleben M *Copyright, Limitations and the Three-Step Test* (2004) Kluwer Law International

Shemtov N *Beyond the Code: Protection of Non-Textual Features of Software* (2017) Oxford University Press

Stokes S *Digital Copyright: Law and Practice* 3ed (2009) Oxford University Press

Van Caenegem W Intellectual Property Law and Innovation (2007) Cambridge University Press

Van der Merwe D *Computers and the Law* 2ed (2000) Juta

Van der Merwe D, Roos A, Pistorius T, Eiselen G & Nel S *Information and Communications Technology Law* 2ed (2016) LexisNexis

Van Genugten W & Meijknecht A (eds.) *Harnessing Intellectual Property Rights for Development Objectives - The Double Role of IPRs in the Context of Facilitating MDGs Nos. 1 and 6* (2011) Wolf Legal Publishers

Van Hoecke M *Legal Doctrine: Which Method(s) for What Kind of Discipline?* (2011) Hart Publishing

Wang J *Conceptualizing Copyright Exceptions in China and South Africa* (2018) Springer

Cases

Australia

Apple Computer Inc and Another v Computer Edge Pty Ltd and Suss [1984] F.S.R. 481

Apple Computer Inc and Another v Computer Edge Pty Ltd and Suss [1984] F.S.R. 246

Computer Edge Pty Limited and Another v Apple Computer Inc and Another [1986] F.S.R. 537

European Union

Infopaq International v Danske Dagblades Forening 2012 EUECJ C-302/10

Nintendo Company Ltd v PC Box Srl Case C-355/12 [2014] E.C.D.R. 6

SAS Institute Inc v World Programming Ltd CJEU Case C-406/10 [2012] R.P.C. 31 [2013] Bus. L.R. 941

South Africa

Apple Computer v Rosy t/a SA Commodity Brokers (Pty) Ltd and Another 1984 JOC (13) 134 (D)

Bosal Afrika (Pty) Ltd v Grapnel (Pty) Ltd 1985 (4) SA 882 (C)

Cavendish Textiles Limited v Manmark Pty Ltd 1984 115 JOC (T)

Certification of the Constitution of the Republic of South Africa, 1996 (CCT 23/96) [1996] ZACC 26; 1996 (4) SA 744 (CC)

Dexion Europe Ltd v Universal Storage Systems (Pty) Ltd 2003 (1) SA 31 (SCA)

Econostat (Pty) Ltd v Lambrecht and Another 1983 89 JOC (W)

First National Bank of SA Limited t/a Wesbank v Commissioner for the South African Revenue Services and Another; First National Bank of SA Limited t/a Wesbank v Minister of Finance (CCT19/01) [2002] ZACC 5; 2002 (4) SA 768

Galago Publishers (Pty) Ltd and another v Erasmus 1989 (1) SA 276 (A)

Haupt t/a Soft Copy v Brewers Marketing Intelligence (Pty) Ltd and Others 2004 BIP 207 (C)

Haupt t/a Softcopy v Brewers Marketing Intelligence (Pty) Ltd and Others 2006 4 SA 458 (SCA)

King v South African Weather Service 2007 BIP 323 (T)

Lacfin (Pty) Ltd v Johannes Nicholaas Le Roux and Four Others 2000 BIP 190 (O)

Laugh It Off Promotions CC v South African Breweries International (Finance) BV t/a Sabmark International and Another 2006 (1) SA 144 (CC)

Lawyers for Human Rights v Minister of Home Affairs and Others (CCT38/16) [2017] ZACC 22; 2017 (5) SA 480 (CC)

Logistics Network (Pty) Ltd v Hard & Software Systems CC and Others 1999 BIP 278 (C)

Makate v Vodacom (Pty) Ltd 2016 (4) SA 121 (CC)

Moneyweb (Pty) Limited v Media 24 Limited and Another 2016 (4) SA 591 (GJ)

Natal Picture Framing Company Ltd v Levin 1920 WLD 35

National Coalition for Gay and Lesbian Equality and Others v Minister of Home Affairs and Others [1999] ZACC 17; 2000 (2) SA 1

National Treasury and Others v Opposition to Urban Tolling Alliance and Others (CCT 38/12) [2012] ZACC 18; 2012 (6) SA 223 (CC)

Prism Holdings Ltd and Another v J Liversage And Others 2001 BIP 114 (W)

Rapid Phase Entertainment CC and others v SABC 1997 JOL 393 (W)

South African Broadcasting Corporation SOC Ltd v Via Vollenhoven and Appollis Independent CC and Others [2016] 4 All SA 623

Technical Information Systems (Pty) Ltd v Marconi Communications (Pty) Ltd 2007 1047 JOC (W)

The Prime Software v Commercial and Industrial Computer Systems 1996 632 JOC (W)

United Kingdom

Apple Computer Inc and Another v Computer Edge Pty Ltd and Suss [1984] F.S.R. 481

Apple Computer Inc and Another v Computer Edge Pty Ltd and Suss [1984] F.S.R. 246

British Leyland v Armstrong [1986] A.C. 577

Cantor Fitzgerald International v Tradition (UK) Ltd [2000] R.P.C. 95

Computer Edge Pty Limited and Another v Apple Computer Inc and Another [1986] F.S.R. 537

Forensic Telecommunications Services Ltd v Chief Constable of West Yorkshire [2011] EWHC 2892 (Ch)

Gates and Others v Swift and Others [1982] R.P.C. 339

IBCOS Computers Ltd. and Another v Barclays Mercantile Highland Finance Ltd. and Others [1994] F.S.R. 275

John Richardson Computers Limited v Flanders and Another [1993] F.S.R. 497

Kabushiki Kaisha Sony Computer Entertainment Inc. and Others v Ball and Others [2004] EWHC 1738

Mars UK Ltd v Teknowledge Ltd [2000] E.C.D.R. 99

Navitaire Inc v easyJet Airline Co Ltd [2004] EWHC 1725 (Ch)

Nintendo Company Ltd v Playables Ltd, Wai Dat Chan [2010] EWHC 1932 (Ch)

SAS Institute Inc v World Programming Ltd [2013] EWCA Civ 1482

SAS Institute Inc v World Programming Ltd CJEU Case C-406/10 [2012] R.P.C. 31 [2013] Bus. L.R. 941

Sid & Marty Krofft Television Productions Inc v McDonald's Corporation 1977 562 F.2d 1157 (9th Cir. 1977)

Sega Enterprises Limited v Richards and Another [1983] F.S.R. 73

Thrustcode Limited and Another v W.W. Computing Limited [1983] F.S.R. 502

Total Information Processing Systems Limited v Daman Limited [1992] F.S.R. 171

United States of America

321 Studios v Metro Goldwyn Mayer Studios Inc (2004) 307 F. Supp. 2d 1085

Acuff-Rose Music Inc v Campbell 972 F.2d 1429

Amicus Brief: Google LLC v Oracle America Inc No 18-956 2019 U.S. S. CT. BRIEFS LEXIS 4933

Apple Computer Inc v Formula International Inc (1983) 562 F. Supp. 775

Apple Computer Inc v Franklin Computer Corporation (1982) 545 F Supp 812

Apple Computer Inc v Franklin Computer Corporation (1983) 714 F 2d 1240

Apple Computer v Formula International Inc. (1983) 562 F. Supp. 775

Apple Computer Inc v Franklin Computer Corporation 714 F.2d 1240, 70 A.L.R.Fed. 153, 219 USPQ 113 (3rd Cir. 1983)

Arnstein v. Porter 1946 154 F.2d 464 (2d Cir. 1946)

Atari Games Corporation and Tengen Inc v Nintendo of America Inc and Nintendo Company Ltd (1991) U.S. Dist. LEXIS 5519

Atari Games Corporation and Tengen Inc v Nintendo of America Inc and Nintendo Company Ltd (1992) 1992 U.S. App. LEXIS 21817; 975 F.2d 832

Atari Games Corporation and Tengen Inc v Nintendo of America Inc and Nintendo Company Ltd (1993) 1993 U.S. Dist. LEXIS 8183

Atari Inc v Amusement World Inc No. Y-81-803 (D. Md. Nov. 27, 1981)

Atari Inc v North American Philips Consumer Electronics Corp. 672 F.2d 607 (7th Cir. 1982)

Baker v Selden 1879 101 US 99

Bonito Boats Inc v Thunder Craft Boats Inc 1986 489 U.S. 141

Campbell v Acuff-Rose Music Inc (1994) 510 U.S. 569

Computer Associates International Inc v Altai Inc 1992 982 F.2d 693

Data Cash Systems Inc v JS&A Group Inc 480 F. Supp. 1063 (N.D. Ill. 1979)

Data Cash Systems Inc v JS&A Group Inc. 628 F.2d 1038 (7th Cir. 1980)

Davidson Associates v Jung (2005) 422 F.3d 630

Dreamcatcher Software Development LLC v Pop Warner Little Scholars Inc (2004) 298 F. Supp. 2d 276

EF Johnson Company v Uniden Corporation of America (1985) 623 F. Supp. 1485

Engineering Dynamics Inc v Structural Software Inc (1994) 26 F.3d 1335

Engineering Dynamics Inc. v Structural Software Inc (1995) 46 F.3d 408

Feist Publications Inc v Rural Telephone Service Company Inc (1991) 111 S. Ct. 1282

GCA Corporation v Chance (1982) 217 U.S.P.Q. (BNA) 718

Google LLC v Oracle America Inc - On Petition For A Writ Of Certiorari To The United States Court Of Appeals For The Federal Circuit Unfiled

Google LLC v Oracle America Inc 2019 U.S. LEXIS 3060

Harper & Row Publishers Inc v Nation Enterprises (1985) 471 U.S. 539

Kepner-Tregoe Inc v Leadership Software Inc (1994) 12 F.3d 527

Lexmark International Inc v Static Control Components Inc (2004) 387 F.3d 522

Lotus Development Corp v Borland International Inc 1995 49 F 3d 807

Midway Manufacturing Company v Artic International Inc No. 80 C 5863 (N.D. Ill. March 10, 1982)

Midway Manufacturing Company v Omni Video Games Inc 668 F.2d 70 (1st Cir. 1981)

Midway Manufacturing Company v Dirkschneider 543 F. Supp. 466 (D. Neb. 1981)

Nichols v Universal Pictures Company 1930 45 F 2d 119

North American Clearing Inc v Brokerage Computer Systems (2009) 666 F. Supp. 2d 1299

Oracle America Inc v Google Inc (2012) 872 F. Supp. 2d 974

Oracle America Inc v Google Inc (2014) 750 F.3d 1339

Oracle America Inc v Google Inc (2014) 750 F.3d 1339

Oracle America Inc v Google Inc 2012 U.S. Dist. LEXIS 66417

Oracle America Inc v Google Inc No. 3:10-cv-3561 (N.D. Cal. June 20, 2012), ECF No. 1211.

Oracle America Inc v Google Inc No. C10-0356 WHA (26 May 2016)

Oracle America Inc v Google LLC (2018) 886 F.3d 1179

Plains Cotton Cooperative Association v Goodpasture Computer Service Inc (1987) 807 F.2d 1256

Roth Greeting Cards v United Card Company 1970 429 F.2d at 1110 (9th Cir. 1970)

SAS Institute Inc v S&H Computer Systems Inc 605 F. Supp 816 (M.D. Tenn. 1985)

SAS Institute Inc v World Programming Ltd (2017) 874 F.3d 370

Sid & Marty Krofft Television Productions Inc v McDonald's Corporation 1977 562 F.2d 1157 (9th Cir. 1977)

Sega Enterprises Ltd v Accolade 1992 U.S. Dist. LEXIS 16132

Sega Enterprises Ltd v Accolade Inc (1992) 785 F. Supp. 1392

Sega Enterprises Ltd v Accolade Inc 1993 U.S. App. LEXIS 78

Sega Enterprises Ltd v Accolade Inc 1992 U.S. App. LEXIS 24121

Sega Enterprises Ltd v Accolade Inc (1992) 977 F.2d 1510

Sony Computer Entertainment America Inc v Divineo Inc et al (2006) 457 F. Supp. 2d 957

Sony Computer Entertainment Inc v Connectix Corporation (1999) 48 F. Supp. 2d 1212

Sony Computer Entertainment Inc v Connectix Corporation (2000) 203 F.3d 596

Sony Corporation of America v Universal City Studios Inc (1984) 464 U.S. 417

Southco Inc v Kanebridge Corporation 2004 390 F.3d 276

Stern Electronics, Inc. v. Kaufman 669 F.2d 852 (2d Cir. 1982)

Synercom Technology Inc. v University Computing Company and Engineering Dynamics Inc. 1978 462 F. Supp. 1003

Syntek Semiconductor Company v Microchip Technology Inc (2002) 307 F.3d 775

Tandy Corp v Personal Micro Computers Inc 524 F. Supp. 171 (N.D. Cal. 1981)

Texas Instruments Inc v Hyundai Electronics Industries Company (1999) 190 F.R.D. 413

United States of America v Elcom Ltd (2002) 203 F. Supp. 2d 1111

Universal City Studios v Corley (2001) 273 F.3d 429

Universal City Studios v Reimerdes (2000) 111 F. Supp. 2d 294

Vault Corporation v Quaid Software Ltd (1988) 847 F.2d 255

Veracode Inc v Appthority Inc 2013 U.S. Dist. LEXIS 146059

Versata Software Inc v Infosys Technologies Ltd 2013 U.S. Dist. LEXIS 186540

Walt Disney Productions v Filmation Associates (1986) 628 F.Supp. 871

Whelan Associates Inc v Jaslow Dental Laboratory Inc (1986) 797 F.2d 1222

Whelan Associates v Jaslow Dental Laboratory Inc (1985) 609 F. Supp. 1307

Williams Electronics Inc v Artic International Inc 1981 U.S. Dist. LEXIS 17856

*Williams Electronics Inc v Artic International Inc.*685 F.2d 870, 215 USPQ 405 (3rd Cir. 1982)

Williams v Arndt (1985) 626 F. Supp. 571

Journal articles

Bainbridge D I "Computers and Copyright" 1987 *Modern Law Review* 50 (2) 202

Bainbridge D I "The Copyright (Computer Software) Amendment Act (1985)" 1986 *Modern Law Review* 49 214

Barnard J and Mišćenić E "The role of the courts in the application of consumer protection law: A comparative perspective" 2019 *Journal for Juridical Science* 44 (1) 111

Burk D L "Anticircumvention Misuse" 2003 *UCLA Law Review* 50 1095

Carty H and Hidkinson K "Copyright, Designs and Patents Act 1988" 1989 *Modern Law Review* 52 (3) 369

Cohen J E "Reverse Engineering and The Rise of Electronic Vigilantism: Intellectual Property Implications of "Lock-Out" Programs" 1995 *Southern California Law Review* 68 1091

Dallas S E "Computer Copyright Protection Narrows as Video Game Giants Battle in Atari v Nintendo" 1994 *Denver University Law Review* 71 739

Davidson D M "Protecting Computer Software: A comprehensive analysis" 1983 *Jurimetrics* 23 337

De Beer J, Baarbé J & Ncube C "Evolution of Africa's Intellectual Property Treaty Ratification Landscape" 2018 *The African Journal of Information and Communication* 22 53

De Villiers R "Computer Programs and Copyright: The South African Perspective " 2006 *SALJ* 123 (2)

Dean O H "Copyright Amendment Act, 1983 Protection of industrial and technical works" March 1984 *De Rebus* 114

Dean O H "The great copyright controversy" July 1988 *De Rebus* 469

Dean O H "Copyright Amendment Act, 1992" October 1992 *De Rebus* 755

Dean O H "The case for the recognition of intellectual property in the Bill of Rights" 1997 *Journal of Contemporary Roman Dutch Law* 60 105

Deasy K & Martin A C "Licensing of Intellectual Property Rights Needed for Software Support: A Life Cycle Approach" 1988*Jurimetrics* 28 223

Druey J N "Information Cannot Be Owned: There is More of a Difference than Many Think" 2004 *The Berkman Centre for Internet and Society, Harvard Law School Research Publication* 4 1

Du Bois M "The Appropriate Scope of Property Rights in Registered Designs" 2017 *South African Intellectual Property Law Journal* 5 (1) 34

DuCharme N F "Copyright Protection for Computer Software in Great Britain and the United States: A Comparative Analysis" 1987 *Santa Clara High Technology Law Journal* 3 (2) 257

Dworking G "The Whitford Committee Report on Copyright and Designs Law" 1977 *Modern Law Review* 40 (6) 685

Ebersohn G "Protecting Copyright in Computer Games and Computer Software" 2005 *TSAR* 1

Forero-Pineda C "The impact of stronger intellectual property rights on science and technology in developing countries" 2006 *Research Policy* 35 808

Freedman C D "The Protection of Computer Software in Copyright and the Law of Confidence: Improper Decompilation and Employee-Poaching" 2000 *International Journal of Law and Information Technology* 8 (1) 25

Geiger C, et al. "The Three-Step-Test Revisited: How to Use the Test's Flexibility in National Copyright Law" 2004 *American University International Law Review* 29 (3) 581

Gesmer L T "Developments in the Law of Computer Software Copyright Infringement" 1986 *Jurimetrics* 26 (3) 224

Ginsburg J C "Toward Supranational Copyright Law? The WTO Panel Decision and the "Three-Step Test" for Copyright Exceptions" 2001 *Revue Internationale du Droit d'Auteur* 187 3

Griffiths J "The "Three-Step Test" In European Copyright Law – Problems And Solutions" 2009 *Queen Mary University of London, School of Law Legal Studies Research Paper No 31/2009*

Hale D J "Recent Development: Vault Corp. V. Quaid Software Ltd.: Limits To Copyright Protection For Computer Programs" 1989 *Tulane Law Review* 64 270

Horowitz G B "The Copyright Act of 1978 as it Affects Computer Programs and Data" 1983SALJ 100 301

Jacobs W "The Electronic Communications and Transactions Act: Consumer Protection and Internet Contracts" 2004 *SA Merc LJ* 16

Karas S "Sony Computer Entertainment Inc v Connectix Corp" 2001 *Berkeley Technology Law Journal* 16 33

Karjiker S "Justifications for Copyright: The Moral Justifications" 2013 *South African Intellectual Property Law Journal* (1) 42

Karjiker S "Justifications for Copyright: The Economic Justification" 2014 *South African Intellectual Property Law Journal* (2) 13

Koelman K J "Fixing the three step test" 2006 *European Intellectual Property Review* 28 (8) 407

Kongolo T "Historical Evolution of Copyright Legislation in Africa" 2014 *World Intellectual Property Organization Journal* 5 (2) 163

Kravetz P I "Copyright Protection of Computer Programs" (1998) 80 *Journal of the Patent and Trademark Office Society* 41

Lee D "Reverse Engineering of Computer Programs under the DMCA: Recognizing a 'Fair Access' Defence" 2006 *Marquette Intellectual Property Law Review* 10 (3) 538

Lin D S, Sag M & Laurie R S "Source Code versus Object Code: Patent Implications for the Open Source Community" 2011 *Santa Clara High Technology Law Journal* 18 235

Low P "WTO Decision-Making for the Future" 2011 *World Trade Organization Economic Research and Statistics Division Staff Working Paper ERSD-2011-05*

McManis C R "Intellectual Property Protection And Reverse Engineering Of Computer Programs In The United States And The European Community " 1993 *High Technology Law Journal* 8 (1) 25

Menell P S "Tailoring Legal Protection for Computer Software" 1987 *Stanford Law Review* 39 1329

Menell P S "An epitaph for traditional copyright protection of network features of computer software" 1998 *The Antitrust Bulletin* 43 651

Menell P S "Rise Of The API Copyright Dead?: An Updated Epitaph For Copyright Protection Of Network And Functional Features Of Computer Software" 2018 *Harvard Journal of Law and Technology* 31 305

Miller A R "Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?" 1993 *Harvard Law Review* 106 (5) 977

Miyashita Y "International Protection of Computer Software" 1991 *The John Marshall Journal of Information Technology and Privacy Law* 11 42

Mupangavanhu Y "Fairness a slippery concept: The common law of contract and the Consumer Protection Act 68 of 2008" 2015 *De Jure* 48 (1) 116

Ncube C "Equitable Intellectual Property Protection of Computer Programs In South Africa: Some Proposals For Reform" 2012 *Stell LR* 3 438

Ncube C "Harnessing Intellectual Property for Development: Some Thoughts on an Appropriate Theoretical Framework " 2013 16 *PER/PELJ* 370

Ncube C "Decolonising Intellectual Property Law in Pursuit of Africa's Development" 2016 *World Intellectual Property Organization Journal* 8 (1) 34

Ncube C, Schonwetter, T De Beer J & Oguamanam C "Intellectual Property Rights and Innovation: Assessing Regional Integration in Africa (ARIA VIII) Working Paper 5" 2017 *Open African Innovation Research* (Open AIR)

Nicholson D R "Free Trade Agreements and TRIPS-plus: implications for developing countries in Africa" 2005 *World Library and Information Congress: 71th IFLA General Conference and Council* 195-E

Nicholson D R "Intellectual Property: benefit or burden for Africa?" 2006 *International Federation of Library Associations and Institutions* 32 310

Nimmer D "A Riff on Fair Use in the Digital Millenium Copyright Act" 2000 *University of Pennsylvania Law Review* 148 (3)

Nwauche E S "The public interest in Namibian copyright law" 2009 1 *Namibian Law Journal* 1 (1) 57

Okediji R L "Development in the Information Age - Issues in the Regulation of Intellectual Property Rights, Computer Software and Electronic Commerce" (2004) *Intellectual Property Rights and Sustainable Development* (UNCTAD-ICTSD Project on IPRs and Sustainable Development) 6

OseiTutu J J "Prioritising Human Development in African Intellectual Property Law" 2016 *World Intellectual Property Organization Journal* 8 (1) 23

Palmer L M "Balancing intellectual property rights with public obligations in developing nations: Lessons from Africa" 2006 *Critical Arts* 20 (1) 62

Pistorius T "The copyright protection of computer programs: literary works shunned by the proposed bill" 1991 *De Rebus* 833

Pistorius T and Visser C "The Copyright Amendment Act 125 of 1992 and Computer Programs: A Preliminary Overview" 1992 *SA Merc LJ* 4 346

Pistorius T "Developing Countries and Copyright in the Information Age" 2006 *PELJ* 11 2

Pistorius T "Copyright in the Information Age: The catch-22 of digital technology" 2006 *Critical Arts* 20 (1) 47

Pistorius T and Mwim O S "The impact of digital copyright law and policy on access to knowledge and learning" 2019 *Reading & Writing - Journal of the Reading Association of South Africa* 10 (1) 1

Plaisted D A "Source-to-Source Translation and Software Engineering" 2013 *Journal of Software Engineering and Applications* 6 30

Prestin D "Where to draw the line between reverse engineering and infringement: Sony Computer Entertainment Inc. v Connectix Corp." 2002 *Minnesota Intellectual Property Review* 3 137

Rahmatian A "Neo-Colonial Aspects of Global Intellectual Property Protection" 2009 *Journal of World Intellectual Property* 12 (1) 40

Rajani A H "Davidson & Associates v. Jung. (Re)interpreting Access Controls" 2006 *Berkeley Technology Law Journal* 21 365

Reznick A E "Synercom Technology, Inc. v. University Computing Co.: Copyright Protection for Computer Formats and the Idea/Expression Dichotomy" 1980 *Rutgers Computer & Technology Law Journal* 8 65

Rice D A "Sega and Beyond: A Beacon for Fair Use Analysis…At Least As Far As It Goes" 1994 *University of Dayton Law Review* 19 1131

Rønning H, Thomas P, Tomaselli K G & Teer-Tomaselli R "Intellectual property rights and the political economy of culture" 2006 *Critical Arts* 20 (1) 1

Samuels E "The Idea-Expression Dichotomy in Copyright Law" 1989 *Tennessee Law Review* 56 321

Samuelson P "Fair Use For Computer Programs And Other Copyrightable Works In Digital Form: The Implications Of Sony, Galoob And Sega" 1993 *Journal of intellectual Property Law* 1 49

Samuelson P "Intellectual Property and The Digital Economy: Why the Anti-Circumvention Regulations need to be revised" 1999 *Berkeley Technology Law Journal* 14 519

Samuelson P "A Fresh Look at Tests for Nonliteral Copyright Infringement" 2013 *Northwestern University Law Review* 107 1821

Samuelson P "Functionality and Expression in Computer Programs: Refining the Tests for Software Copyright Infringement" 2016 *Berkeley Technology Law Journal* 31 (2) 1215

Samuelson P and Scotchmer S "The Law and Economics of Reverse Engineering" 2001 *Yale Law Journal* 111 1575

Samuelson P, Davis R, Kapor M D & Reichman JS "A Manifesto Concerning the Legal Protection of Computer Programs" 1994 *Columbia Law Review* 94 2308

Senftleben M "The International Three-Step Test: A Model Provision for EC Fair Use Legislation" 2010 *Journal of Intellectual Property, Information Technology and Electronic Commerce Law* 1 67

Shay R M and Van der Walt A "Constitutional Analysis of Intellectual Property" 2014 *Potchefstroom Electronic Law Journal* 17 (1) 52

Silva A J C "Beyond the Unrealistic Solution for Development Provided by the Appendix of the Berne Convention on Copyright" 2013 *Journal of the Copyright Society of the USA* 60 (4) 581

Simon I "South African Supreme Court Rules on Copyright in Software and Computer-generated Works" 2006 *Journal of Intellectual Property Law & Practice* 1 (11) 696

Snail S L "South African e-consumer law in the Context of the ECT Act (part 1)" 2007 *The Quarterly Law Review for People in Business* 15 (1)

Soobert A M "Legitimizing decompilation of computer software under Copyright law: a square peg in search of a square hole" 1994 *John Marshall Law Review* 28 105

Spivack P G “Does form follow function? The idea/expression dichotomy in copyright protection of computer software” 1988 *UCLA Law Review* 35 723

Staines A “Copyright in Computer Programs: A Hollow Victory” 1983*Modern Law Review* 46 231

Stern R H “Another Look at Copyright Protection of Software: Did The 1980 Act Do Anything for Object Code” (1981) 3 *Computer Law Journal* 1

Tong L “Copyright Protection for Computer Programs in South Africa: Aspects of Sui generis Categorization” 2009 *Journal of World Intellectual Property* 12 (4) 266

Tong L “Authorship of Computer Programs under South African Copyright Law” 2005 *SALJ* 122 (3)

Velasco J “The Copyrightability of Nonliteral Elements of Computer Programs” (1994) 94 *Columbia Law Review* 242

Victor D “An Analysis of an Affirmative Defense for Reverse Engineering Within A System of Legal Protection for Computer Software” 1993 *Southern California Law Review* 66 1705

Visser C “A comparative survey of aspects of the subsistence of copyright in computer software” 1984 *Comparative and International Law Journal of Southern Africa* 17 32

Visser C “Copyright in Works Created in the Course of Employment: The Supreme Court of Appeal Gives Guidance” 2009 *SA Merc LJ* 21 591

Waddington D & Yao B “High-Fidelity C/C++ Code Transformation” (2005) 141 *Electronic Notes in Theoretical Computer Science* 35

Wolfe R “The WTO Single Undertaking as Negotiating Technique and Constitutive Metaphor” 2009 *Journal of International Economic Law* 12 (4) 835

Reports, Public Notices, Working Papers and Policy Documents

CONTU 1978 *Final Report of the National Commission on New Technological Uses of Copyrighted Works*

Department of Trade and Industry *Draft National Policy on Intellectual Property*, 2013 General Notice No. 918 of 2013, Government Gazette (4 September 2013)

Electronic Frontier Foundation 2014 *Unintended Consequences: Sixteen Years under the DMCA*

Gregory S 2008 *Intellectual Property Rights and South Africa's Innovation Future* South African Institute of International Affairs

Judicial House Committee *Copyright Law Revision House Report No. 94-1476, 94th Cong., 2d Sess.*

Librarian of Congress 1980 *83rd Annual Report of the Register of Copyrights*

Librarian of Congress 2018 *Section 1201 Rulemaking: Seventh Triennial Proceeding to Determine Exemptions to the Prohibition on Circumvention*

National Planning Commission, Department: The Presidency 2012 *National Development Plan 2030 Our Future - Make It Work* (NDP)

Pouris A & R Inglezi-Lotz R 2011 *The Economic Contribution of Copyright-Based Industries in South Africa* Published by the Institute for Technological Innovation, University of Pretoria and WIPO in cooperation with the Republic of South Africa Department of Trade and Industry

Rens, A Chetty P, Schonwetter T, Kahn R, Jacobson P & Maistry U 2010 *Report on the South African Open Copyright Review*

US Copyright Office 2017 *Section 1201 of Title 17: A Report of the Register of Copyrights*

Whitford J N K 1977 *Copyright and Designs Law: Report of the Committee to Consider the Law on Copyright and Designs Cmnd. 6732*

WIPO 1971 "Advisory Group of Governmental Experts on the Protection of Computer Programs (Geneva, March 8 to 12,1971)" 7 *Copyright* 35

WIPO 1975 *Advisory Group of Non-Governmental Experts on the Protection of Computer Programs (Second Session) - Legal Protection of Computer Programs AGCP/NGO/ll/4* Geneva

WIPO 1978 "Model Provisions on the Protection of Computer Software" *Copyright* 6.

WIPO 1979 *Expert Group on the Legal Protection of Computer Software (First Session) - Measures to Enhance International Cooperation in the Field of Legal Protection of Computer Software LPCS/1/2* Geneva

WIPO 1983 *Committee of Experts on the Legal Protection of Computer Software (Second Session) - Analysis of the Results of the Survey Concerning the Desirability and Feasibility of a Treaty for the Protection of Computer Software and/or Other Possible Measures in the Field of the Protection or Deposit of Computer Software LPCS/ll/2* Geneva

WIPO 1983 *Committee of Experts on the Legal Protection of Computer Software (Second Session) - Draft Treaty for the Protection of Computer Software LPCS/II/3* Geneva

WIPO 1985 "Group of Experts on the Copyright Aspects of the Protection of Computer Software (Geneva, February 25 to March 1, 1985) - Report" 21 *Copyright* 146.

WIPO 1989 "Committee of Experts on Model Provisions for Legislation in the Field of Copyright (First Session)" 25 *Copyright* 146

WIPO 1990 "Committee of Experts on Model Provisions for Legislation in the Field of Copyright - Draft Model Law on Copyright (Preparatory Document) CE/MPC/III/2" 9 *Copyright* 241

WIPO 1990 "Committee of Experts on Model Provisions for Legislation in the Field of Copyright - Report CE/MPC/III/3" 9 *Copyright* 282

WIPO 1999 *Records of the Diplomatic Conference on Certain Copyright and Neighboring Rights Questions: Summary Minutes of Main Committee I* Geneva

WIPO 2013 *Draft Guidelines on Assessing the Economic, Social and Cultural Impact of Copyright on the Creative Economy*

WIPO *Berne Notification No. 64 Berne Convention for the Protection of Literary and Artistic Works Accession of the Republic of South Africa to the Paris Act (1971) (with the exception of Articles 1 to 21 and the Appendix)*

WIPO Internal Oversight Division 2016 *Evaluation Policy 2016-2020 IOD/EP/2016*

WIPO Internal Oversight Division 2019 *Evaluation Manual IOD/EM/2019*

WIPO Office of Strategic Use of Intellectual Property for Development 2007 *Study on the Economic, Social and Cultural Impact of Intellectual Property in the Creative Industries: Final Report*

WTO Dispute Settlement Panel 2000 *Report of the Panel on United States – Section 110(5) of the US Copyright Act*

Statutory Instruments, Directives, Conventions and Treaties

Agreed Statements Concerning the WIPO Copyright Treaty adopted by the Diplomatic Conference on 20 December 1996

Agreement on Trade Related Aspects of Intellectual Property Rights (TRIPS), Annex 1C of the Marrakesh Agreement Establishing the World Trade Organization 15 April 1994

An Act for the general revision of the Copyright Law, title 17 of the United States Code, and for other purposes Pub.L. 94–553 90 Stat. 2541

An Act to Amend and Consolidate the Acts Representing Copyright (1909) Pub.L. 60–349 35 Stat. 1075

Berne Convention for the Protection of Literary and Artistic Works (Paris Act) as amended on September 28, 1979

Computer Software Copyright Act of 1980 Pub. L. No. 96-517, 94 Stat. 3015, 3028

Constitution of the Republic of South Africa 1996

Consumer Protection Act 68 of 2008

Copyright (Computer Programs) Regulations 1992 No. 3233

Copyright (Computer Software) Amendment Act UK ST 1985 c. 41

Copyright Act of 1909 (An Act to Amend and Consolidate the Acts Representing Copyright) nPub.L. 60–349 35 Stat. 1075

Copyright Act 1956

Copyright Act of 1976 (An Act for the general revision of the Copyright Law, title 17 of the United States Code, and for other purposes) Pub.L. 94–553 90 Stat. 2541

Copyright Amendment Act 56 of 1980

Copyright Amendment Act 66 of 1983

Copyright Amendment Act 13 of 1988

Copyright Amendment Act 125 of 1992

Copyright Amendment Bill [B 13B—2017] (November 2018)

Copyright and Related Rights Regulations 2003 No. 2498

Copyright, Designs and Patents Act 1988 Chapter 48

Council Directive on The Legal Protection of Computer Programs O.J. No. L122 91/250/EEC of 14 May 1991

Designs Act 195 of 1993

Directive of the European Parliament and of the Council on the Harmonisation of Certain Aspects of Copyright and Related Rights in The Information Society Directive 2001/29/EC of 22 May 2011

Directive of the European Parliament and of the Council on the Legal Protection of Computer Programs Directive 2009/24/EC (23 April 2009) *Official Journal of the European Union* 5.5.2009 L111/16

Electronic Communications and Transactions Act 25 of 2002

Exemption to Prohibition on Circumvention of Copyright Protection Systems for Access Control Technologies Federal Register Vol. 83 No. 208, 37 CFR Part 201 [Docket No. 2017–10]

Intellectual Property Rights and Innovation Cmnd. 9117 (1983)

Intellectual Property and Innovation Cmnd. 9712 (1986)

International Covenant on Economic, Social and Cultural Rights Adopted and opened for signature, ratification and accession by General Assembly resolution 2200A (XXI) of 16 December 1966 entry into force 3 January 1976, in accordance with article 27 (ICESCR)

Memorandum on The Copyright Amendment Bill 1992 *Government Gazette* (28 June 1991) No. 13328 Notice 576 of 1991

Reform of The Law Relating to Copyright, Designs and Performers' Protection Cmnd. 8302 (1981)

Semiconductor Chip Protection Act of 1984 17 U.S.C. ch. 9 §901

The Digital Millennium Copyright Act Pub. L. No. 105-304, 112 Stat. 2860, 2887 (Oct. 28, 1998)

Trademark Act of 1946 (Lanham Act) 60 Stat. 427 15 U.S.C. §1051

United States Constitution

Vienna Convention on The Law of Treaties No 18232 (23 May 1969)

WIPO Copyright and Performances and Phonograms Treaties Implementation Act of 1998 Pub. L. No. 105-304 112 Stat. 2861

WIPO Copyright Treaty (WCT) TRT/WCT/001 WO033EN 1996

WIPO Performances and Phonograms Treaty (WPPT) TRT/WPPT/001 1996

Other sources

Burleson D "Have decompilers become evil?" (2019) Burleson Consulting http://www.dba-oracle.com/t_decompilers_evil.htm

Copyright Alliance "Oracle v Google" https://copyrightalliance.org/copyright-law/copyright-cases/oracle-america-v-google/

Dean O H, et al. "Commentary on the Copyright Amendment Bill 2015" http://blogs.sun.ac.za/iplaw/files/2015/08/CIP-Formal-Comments-Copyright-Amendment-Bill-2015-Online1.pdf

Gream M "A brief note on the reverse engineering of protected computer programs from a UK perspective" (2003) http://matthewgream.net/Professional/IntellectualProperty/note_decompile-comp-prog.pdf

Jooste C and Karjiker S "Commentary on the Copyright Amendment Bill 2017" http://blogs.sun.ac.za/iplaw/files/2017/06/CIP-Comments-Copyright-Amendment-Bill-2017.pdf

Karjiker S 2013 *Open-source software and the rationale for copyright protection of computer programs* Thesis: LLD Stellenbosch University

Menell P S and Nimmer D “Google LLC v Oracle America: Brief of Professors Peter S. Menell and David Nimmer As Amici Curiae In Support Of Petitioner” https://www.supremecourt.gov/DocketPDF/18/18956/89422/20190225113504834_37659%20pdf%20Menell.pdf

Nicholson D R “‘Fair use’ in new Copyright Bill benefits everyone” *Daily Maverick* https://www.dailymaverick.co.za/article/2019-09-17-fair-use-in-new-copyright-bill-benefits-everyone/

Myburg A "Opinion: Behind Ramaphosa’s rejection of the Copyright Bill" (2020) *Mail & Guardian* https://mg.co.za/opinion/2020-07-02-behind-ramaphosas-rejection-of-the-copyright-bill/

Nwauche E S 2005 *A Development Oriented Intellectual Property Regime For Africa* Paper presented at the 11th General Assembly of the Council for the Development of Social Science Research for Africa (CODESRIA) Maputo Mozambique 6-10 December 2005

Palmedo M "South Africa’s Copyright Amendment Bill Returned to Parliament for Further Consideration" (2020) *Infojustice* http://infojustice.org/archives /42426

Samuelson P and Crump C “Google LLC v Oracle America Inc: Brief of 65 Intellectual Property Scholars as Amici Curiae in Support of Petitioner” https://www.supremecourt.gov/DocketPDF/18/18956/89474/20190225131314910_IP%20 Scholars%20Amicus%20Brief.pdf

Shay R M 2012 *Users’ Entitlements under the Fair Dealing Exceptions to Copyright* Thesis: LLM Stellenbosch University

The Presidency "President refers Copyright and Performers’ Protection Amendment Bills to Parliament" (2020) *The Presidency of the Republic of South Africa* www.thepresidency.gov.za/press-statements/president-refers-copyright-and-performers’-protection-amendment-bills-parliament

Touretzsky D S "Source vs. Object Code: A False Dichotomy" (2000) http://www.cs.cmu.edu/~dst/DeCSS/object-code.txt

University of Hawaii System "ICS 1111: The Programming Process" http://www2.hawaii.edu/~takebaya/ics111/process_of_programming/process_of_programming.html

Van De Vanter, M L "Preserving the Documentary Structure of Source Code in Language-based Transformation Tools" (2001) *IEEE International Workshop on Source Code Analysis and Manipulation (SCAM 2001),* 10 November 2001, Florence, Italy